National Aeronautics and Space Administration

Goddard Earth Science Data Information and Services Center

OCO-2 Data Product User's Guide, Pre-Launch, Simulated Data Files

Revision Date: Revision B, June 16, 2014

Jet Propulsion Laboratory
Pasadena, California

Goddard Space Flight Center
Greenbelt, Maryland

Preparers

Gregory Osterman, JPL
OCO-2 Science Validation Lead

Annmarie Eldering, JPL
OCO-2 Deputy Project Scientist

Charles Avis, JPL
ACOS, OCO-2 Science Data Operations System

Brian Chafin, JPL
ACOS, OCO-2 Science Data Operations System

Christopher O'Dell, CSU
ACOS, OCO-2 Algorithm Team

Christian Frankenberg, JPL
ACOS, OCO-2 Algorithm Team

Brendan Fisher, JPL
ACOS, OCO-2 Validation Team

Debra Wunch, Caltech
ACOS, OCO-2 Validation Team

Revision History

Revision Date	Changes	Author
A - 11 April 2014	Initial release, pre-launch release of simulated data	G. Osterman and team
B - 16 June 2014	Update to pre-release, correcting MaxMS values	G. Osterman and team

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Document Overview

This document provides a brief overview of the Orbiting Carbon Observatory-2 (OCO-2) mission and then discusses the content of the publically available OCO-2 data products. Section 2 provides an overview of the OCO-2 mission. Section 3 defines the naming conventions that are used throughout the data products. Section 4 discusses the key data fields in the L2 (Standard and Diagnostic) product. Section 5 discusses product characteristics and key data fields and provides recommendations for data analysis. Section 6 describes the IMAP-DOAS data product key fields. Section 7 focuses on the calibrated radiances found in the L1bSc product. Section 8 provides full tables of all of the fields in the data products. Section 9 lists tools to view and search the data products. Section 10 lists contact information for questions or issues with the OCO-2 data. Section 11 lists acknowledgements and relevant publications, and the last section lists the abbreviations and acronyms used in this document.

Please note that this is a preliminary document only. This initial release of sample OCO-2 data is designed to provide users with information on data formats and volumes. This document will be updated once actual OCO-2 data is available and with new data releases in the future. In addition, there are algorithm theoretical basis documents (ATBD) that discuss the physics and algorithm details.

1.2 Data Usage Policy

These data have been produced by the OCO-2 project, and are provided freely to the public. In order to improve our product and have continued support for this work, we need user feedback and also have users acknowledge data usage. Therefore, we request that when publishing using OCO-2 data, please acknowledge NASA and the OCO-2 project.

- Include OCO-2 as a keyword to facilitate subsequent searches of bibliographic databases if it is a significant part of the publication
- Include a bibliographic citation for OCO-2 data. The most relevant citations currently are Wunch et al (2011a and 2011b), O'Dell et al (2011), Crisp et al (2012), and Frankenberg et al (2014).
- Include the following acknowledgements: "These data were produced by the OCO-2 project at the Jet Propulsion Laboratory, California Institute of Technology, and obtained from the ACOS/OCO-2 data archive maintained at the NASA Goddard Earth Science Data and Information Services Center."
- We recommend sending courtesy copies of publications to the OCO-2 Project Scientist, Michael.R.Gunson@jpl.nasa.gov and Deputy Project Scientist, Annmarie.Eldering@jpl.nasa.gov.

2 Mission Overview

The Orbiting Carbon Observatory is the first NASA mission designed to collect space-based measurements of atmospheric carbon dioxide with the precision, resolution, and coverage needed to characterize the processes controlling its buildup in the atmosphere [Crisp et al. 2004; 2008]. After a launch mishap, which prevented the original OCO mission from reaching orbit, the Orbiting Carbon Observatory-2 mission was formulated to meet the original OCO objectives.

Fossil fuel combustion and other human activities are now emitting more than 30 billion tons of carbon dioxide into the atmosphere every year. Atmospheric CO_2 measurements currently being collected by a global network of surface stations indicate that less than half of the CO_2 is accumulating in the atmosphere. The remainder is apparently being absorbed by CO_2 "sinks" in the ocean and the terrestrial biosphere [c.f. Canadell et al. 2007]. While the existing surface greenhouse gas monitoring network has expanded continuously over the past 50 years and now provides the accuracy and coverage needed to quantify the abundance of this gas on global scales, it still lacks the spatial and temporal resolution and coverage needed to identify and quantify CO_2 sources and sinks on regional scales or to quantify emissions from discrete point sources.

One way to improve the coverage and resolution of these measurements is to collect spatially resolved, global measurements of the column-averaged CO_2 dry air mole fraction, X_{CO2}, from space [Rayner and O'Brien 2001]. Although natural processes and human emissions can change the atmospheric CO_2 mixing ratio by as much as 8% near the surface (>30 ppm out of the ~385 ppm background), the amplitude of these variations decreases rapidly with altitude, such that X_{CO2} variations rarely exceed 2% (8 ppm) on regional to global scales. East-west variations are typically no larger than 0.3 to 0.5%. Because of this, modeling studies show that space based measurements of X_{CO2} can substantially improve our understanding of surface fluxes only if they have the accuracy, precision, coverage, spatial resolution, and temporal sampling needed to describe X_{CO2} variations with amplitudes no larger than 0.3 to 0.5 % (1 to 2 ppm) on scales ranging from < 100 km over continents, to ~1000 km over the ocean [Rayner and O'Brien 2001].

Systematic biases with amplitudes larger than 0.3% on spatial scales of 100 to 1000 km will introduce spurious X_{CO2} gradients that would be indistinguishable from those produced by true CO_2 sources or sinks. Absolute X_{CO2} accuracies better than 0.3% on these scales are therefore essential for retrieving CO_2 fluxes. Truly global biases are less of a concern because they will not introduce spurious X_{CO2} gradients. However, such biases could compromise validation of the space-based measurements against other standards, such as the World Meteorological Organization (WMO) standard for atmospheric CO_2.

Space-based measurements of X_{CO2} are likely to make their most significant contributions to our understanding of the carbon cycle over the ocean and over tropical land masses, because these regions are poorly sampled by the existing ground-based network. X_{CO2} estimates over the ocean are needed to quantify their large natural CO_2 sources and sinks and to facilitate the tracking of CO_2 emissions transported over the ocean by the prevailing winds. X_{CO2} measurements must also be collected over nearly the full range of latitudes on the sunlit hemisphere to avoid uncertainties introduced by the transport of air in and out of the field of regard.

To resolve CO_2 fluxes on spatial scales ranging from <100 to ~1000 km, data must be collected at higher resolution to discriminate natural sinks from nearby sources. A small sampling footprint also helps to ensure that some cloud-free soundings can be obtained even in partially cloudy regions, since the probability of measuring a cloud free scene is inversely proportional to footprint size. A small sounding footprint is also needed to quantify CO_2 emissions from discrete point sources, such as individual power plants or cities because the minimum detection limit (measured in kg of CO_2) associated with a given concentration change (e.g., a 1 ppm variation in X_{CO2}) is inversely proportional to the area of the footprint.

The natural processes responsible for the uptake and release of CO_2 are driven primarily by photosynthesis and respiration on land and by the solubility of CO_2 in the ocean. The efficiency of these natural processes varies on diurnal, seasonal and interannual time scales. CO_2 emissions from human activities also vary on these time scales. Existing ground-based measurements indicate that while diurnal CO_2 variations in the vicinity of local sources and sinks can be large (>10 ppm), these variations are confined near the surface, and rarely contribute to X_{CO2} variations larger than 0.3%. While these small differences would be difficult to detect from space, X_{CO2} should be estimated from measurements acquired at the same, fixed time of day everywhere on Earth to avoid introducing a spatially varying diurnal bias. Global measurements are needed at on semi-monthly intervals over a complete annual cycle to identify changes in the natural and human contributions to atmospheric CO_2 over the seasonal cycle. More than one seasonal cycle must be observed to resolve the relative contributions of seasonal and interannual variability to the atmospheric CO_2 buildup.

2.1 Measurement Approach

To meet these objectives, OCO-2 employs a dedicated spacecraft with a single instrument that will be launched into a near-polar orbit on an expendable launch vehicle [Crisp et al. 2007]. The Observatory will fly in a loose formation with the Earth Observing System Afternoon Constellation (EOS A-Train). This 705 km altitude, sun synchronous orbit follows the World Reference System-2 (WRS-2) ground track, yielding 233 orbits over its 16-day ground track repeat cycle. The orbit's 1:30 PM mean local time is well suited for acquiring observations of the absorption of reflected sunlight by CO_2 and O_2 because the sun is high, maximizing the available signal. It also facilitates coordinated calibration and validation campaigns with other A-Train instruments, and synergistic use of OCO-2 data with that from other A-Train platforms.

The OCO-2 instrument incorporates three high-resolution spectrometers that make coincident measurements of reflected sunlight in the near-infrared CO2 near 1.61 and 2.06 µm and in molecular oxygen (O_2) A-Band at 0.76 µm [Crisp et al. 2007]. Simultaneous, co-boresighted measurements from these 3 spectral regions are combined to define a single "sounding." Each sounding is analyzed with remote sensing retrieval algorithms to produce an estimate of X_{CO2} for the atmospheric path between the sun, the reflecting surface and the OCO-2 instrument. Because the dry air mole fraction of O2 is well known and essentially constant, measurements of O_2 A-band absorption provide direct constraints on the surface pressure and uncertainties in the atmospheric optical path length introduced by cloud and aerosol scattering and pointing errors. Measurements of absorption by the weak and strong CO_2 absorption bands near 1.61 and 2.06 µm, respectively, provide information about both the CO_2 column abundance and the wavelength-dependent scattering by aerosols along the same optical path.

The OCO-2 instrument must measure CO_2 and O_2 absorption with adequate precision to yield X_{CO2} estimates with a precision better than 0.3% on spatial scales smaller than 100 km over continents and 1000 km over the ocean over more than 90% of range of latitudes on the on the sunlit hemisphere of the Earth. To meet these objectives, the instrument must have a high sensitivity and a high signal-to-noise ratio (SNR) over a wide dynamic range [Crisp et al. 2008]. A high spectral resolving power ($\lambda/\delta\lambda > 20{,}000$) is needed to resolve the CO_2 and O_2 lines from the adjacent continuum to maximize the sensitivity to small (< 0.3%) variations in X_{CO2}.

Measurements across the entire O_2 or CO_2 band are needed at high SNR because a 0.3% variation in X_{CO2} must be inferred from substantially smaller variations in O_2 and CO_2 absorption strength. The retrieval algorithm must then perform a least squares fit to dozens of lines within each band to yield X_{CO2} retrievals with precisions near 0.3%. A wide dynamic range is needed because the contrast between line cores and the adjacent continuum can exceed 100:1, and because the signal level depends on the intensity of the sunlight reflected from the surface, which decreases with increasing solar zenith angle (latitude) and decreasing surface reflectance.

The OCO-2 instrument will collect 8 soundings over its 0.8-degree wide swath every 0.333 seconds, yielding surface footprints with along track dimensions < 2.25 km and cross-track dimensions that vary from 0.1 to 1.3 km at nadir [Crisp et al., 2008]. The high spatial resolution will facilitate the discrimination of natural sinks from nearby sources and enhance the coverage by increasing the probability of collecting some cloud free soundings even in partially cloudy conditions. The high sampling rate will provide dozens to hundreds of samples over downtrack distances that are small compared to those that characterize the spatial variability of X_{CO2} over continents (<100 km) and ocean (1000 km), even when less than 10% of samples are sufficiently cloud free to yield full-column X_{CO2} measurements. X_{CO2} estimates from these soundings can therefore be averaged to improve the precision. While the rapid downtrack sampling yields high spatial resolution along the orbit tracks, the east-west (left/right of ground track) resolution is largely determined by the distance between orbit tracks. The 98.8 minute orbit period, yields ~14.56 orbits each day that are separated by ~24.7° of longitude. The orbit track spacing decreases to ~13° after 2 days, and to 1.5° after a full 16-day repeat cycle.

To ensure their accuracy, the space based X_{CO2} estimates are validated through comparisons with near-simultaneous measurements of X_{CO2} acquired by ground-based Fourier Transform Spectrometers in the Total Carbon Column Observing Network (TCCON) [Washenfelder et al. 2006; Wunch et al. 2010]. This network currently includes over a dozen stations, distributed over a range of latitudes ranging from Lauder New Zealand to Ny Alesund, Norway, and is continuing to add new facilities. To relate TCCON measurements to the WMO CO_2 standard, aircraft observations have been collected over several stations, using in situ CO_2 measurement approaches used to define that standard. OCO-2 will target a TCCON site as often as once each day, acquiring thousands of measurements as it flies overhead. These measurements will be analyzed to reduce biases below 0.1% (0.3 ppm) at these sites. The space-based X_{CO2} estimates will be further validated through comparisons with CO_2 and surface pressure measurements from ground based sites with the aid of data assimilation models to provide a more complete global assessment of measurement accuracy.

2.2 Instrument Characteristics

The Observatory carries and points a single instrument that incorporates three co-boresighted, long-slit, imaging grating spectrometers optimized for the O_2 A-band at 0.765 μm and the CO_2 bands at 1.61 and 2.06 μm (Figure 2-1) [Crisp et al. 2007; Crisp 2008]. The instrument mass is ~140 kg, and its average power consumption is ~100 Watts. The 3 spectrometers use similar optical designs and are integrated into a common structure to improve system rigidity and thermal stability. They share a common housing and a common F/1.8 Cassegrain telescope. The light path is illustrated in Figure 2-1. Light entering the telescope is focused at a field stop and then recollimated before entering a relay optics assembly. There, it is directed to one of the three spectrometers by a dichroic beam splitter, and then transmitted through a narrowband pre-disperser filter. The pre-disperser filter for each spectral range transmits light with wavelengths within ~±1% of the central wavelength of the CO_2 or O_2 band of interest and rejects the rest. The light is then refocused on the spectrometer slits by a reverse Newtonian telescope. Each

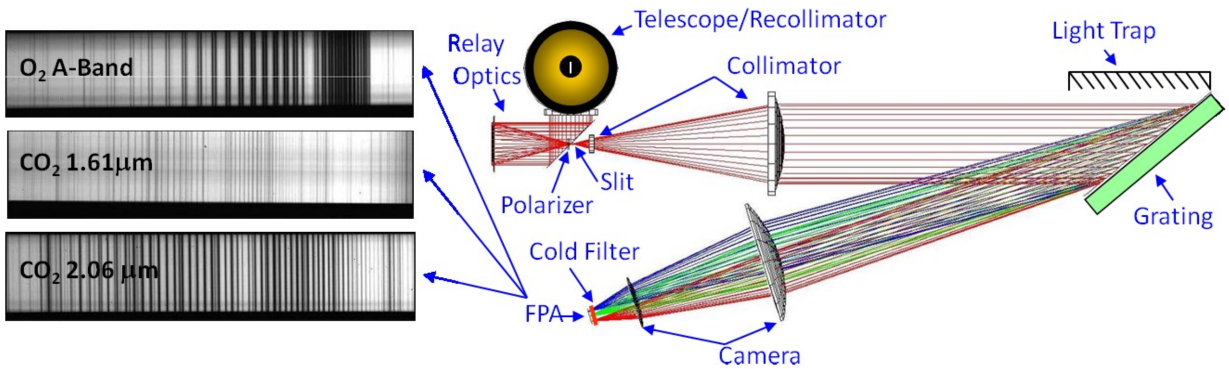

Figure 2-1. *The OCO-2 instrument showing the major optical components and optical path (right) and images of spectra recorded by the FPA in the 3 spectral channels (left).*

spectrometer slit is ~3 mm long and ~25 µm wide. These long, narrow slits are aligned to produce co-boresighted fields of view that are ~0.0001 radians wide by ~0.0146 radians long. Because the diffraction gratings efficiently disperse light that is polarized in the direction parallel to the slit, a polarizer was included in front of the slit to reject the unwanted polarization before it enters the spectrometer, where it could contribute to the scattered light background.

Once the light enters a spectrometer slit, it is collimated by a 2-element refractive collimator, dispersed by a reflective planar holographic diffraction grating, and then focused by a 2-element camera lens on a 2-dimensional focal plane array (FPA), after traversing a second, narrowband filter. The narrowband filter just above the FPA is cooled to ~180K to reject thermal emission from the instrument.

The spectral range and resolving power of each channel includes the complete molecular absorption band as well as some nearby continuum to provide constraints on the optical properties of the surface and aerosols as well as absorbing gases. To meet these requirements, the O_2 A-band channel covers 0.758 to 0.772 µm with a resolving power of >17,000, while the 1.61 and 2.06 µm CO_2 channel cover 1.594 to 1.619 µm and 2.042 to 2.082 µm, respectively with a resolving power > 20,000 [Crisp et al. 2007; Crisp 2008].

The gratings disperse the light onto the FPAs in the direction orthogonal to the long dimension of the slit (Figure 2-2). The field of view is resolved spatially along the slit. The FPAs are 1024 x 1024 pixel arrays with 18 µm by 18 µm pixels that have a 100% fill factor (i.e., there are no spatial or spectral gaps between the pixels). The slit is imaged on the FPA, which samples its full-width at half maximum (FWHM) with 2 to 3 pixels. The quantum efficiency of the FPAs is between 75 and 90%, and the read noise is < 20 electrons/pixel/exposure. The FPA temperatures are maintained at <120 K by a pulse-tube cryocooler that is thermally coupled to an external radiator though variable conductance heat pipes. At this temperature, thermal noise from the FPAs is negligible during the short exposure time (0.333 seconds). The optical bench is maintained at -5 °C by a thermal radiative shroud that is coupled to an external radiator by variable conductance heat pipes.

The spectrum produced by each channel is dispersed to illuminate all 1024 pixels in spectral dimension on each FPA. The length of the slit limits spatial field of view to only ~190 pixels in the spatial dimension (Figure 2-2a). OCO-2 soundings use an along-slit field of view is defined by ~160 of these 190 pixels. In normal science operations, the FPAs are continuously read out at 3 Hz. To reduce the downlink data rate and increase the SNR, 20 adjacent pixels in the FPA dimension parallel to the slit (i.e., the "Spatial Direction" in Figure 2-2a) are summed on board

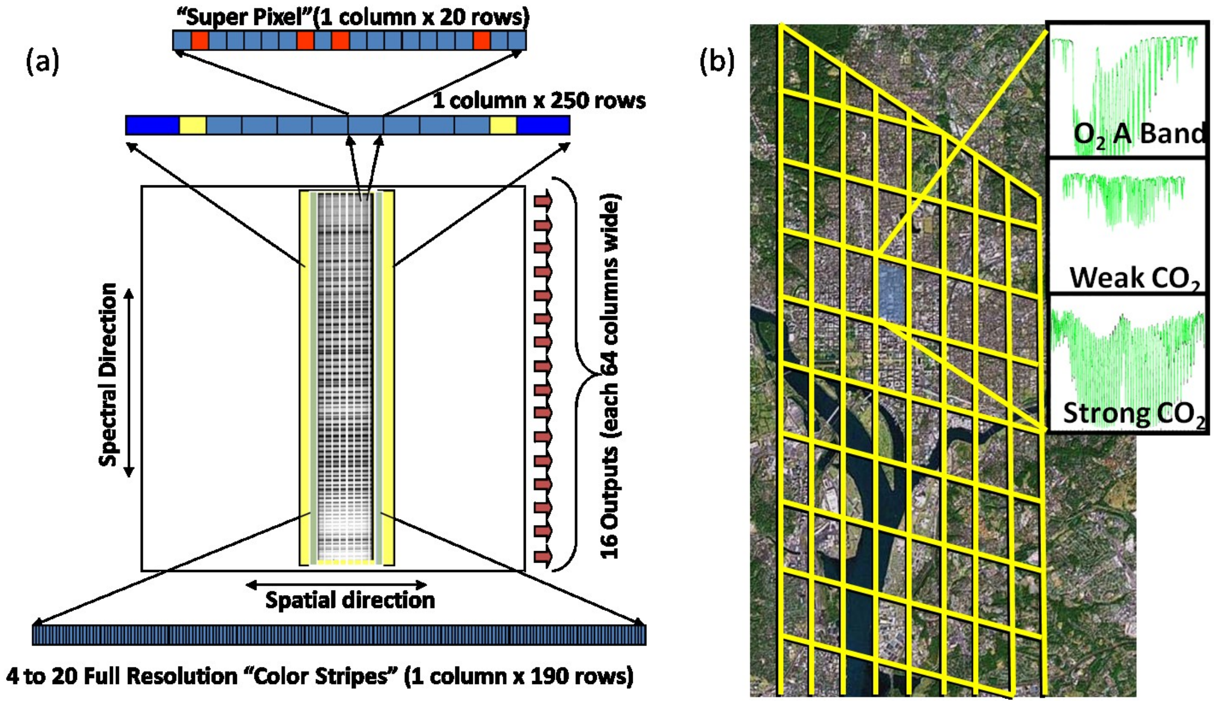

Figure 2-2. *(a) The illumination and readout scheme used for the OCO-2 Focal Plain Arrays. (b) Spatial layout of 8 cross-track footprints for nadir observations over Washington DC.*

to produce up to 8 spatially-averaged spectra (Figure 2-2b). The along-slit angular field of view of each of these spatially-averaged "super-pixels: is ~1.8 mrad (0.1° or ~1.3 km at nadir from a 705 km orbit). The angular width of the narrow dimension of the slit is only 0.14 mrad, but the telescope focus was purposely softened to increase the effective full width at half maximum of each slit to ~0.6 mrad to simplify the boresight alignment among the 3 spectrometer slits.

In addition to the 8 spatially binned, 1024-element spectra, each spectrometer also returns 4 to 20 spectral samples without on-board spatial binning to provide the full along-slit spatial resolution. Each of these full-resolution "color stripes" covers a 220 pixel wide region of the FPA that includes the full length of the slit (190 pixels) as well as a few pixels beyond the ends of the slit (Figure 2-2). These full-spatial-resolution color stripes are used to detect spatial variability within each of the spatially summed super pixels and to monitor the thermal emission and scattered light within the instrument.

2.2.1 Observing modes

The spacecraft bus orients the instrument to collect science observations in Nadir, Glint, and Target modes (Figure 2-3) [Crisp et al. 2007; Crisp 2008]. In Nadir mode, the satellite points the instrument aperture to the local nadir, so that data can be collected along the ground track just below the spacecraft. In Glint mode, the attitude and control system (ACS) is programmed to point the instrument aperture toward the bright "glint" spot. In Target mode, the ACS points the instrument's aperture at specific stationary surface targets as the satellite flies overhead. To ensure that the target is not missed, the ACS superimposes a small amplitude sinusoidal oscillation (±0.23° about the spacecraft y axis) in the direction perpendicular to the long dimension of the spectrometer slit, to scan the slits over a region centered on the nominal target location Figure 2-3c. This "Target scan", combined with the instruments 0.8° wide field of view, creates a 0.46° by 0.8° viewing box around the target. The period of this sinusoidal oscillation

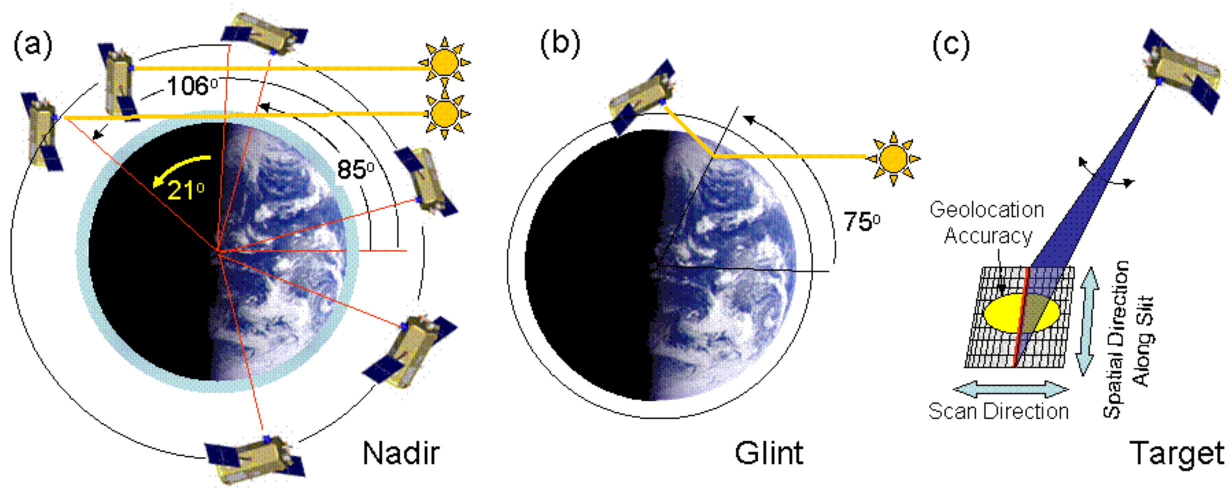

Figure 2-3. *Nadir, Glint, and Target observations. (a) Nadir observations are acquired over the sunlit hemisphere at latitudes where the surface solar zenith angle is <85°. On all orbits except downlink orbits, as the Observatory passes over the northern terminator, it pitches up to point the instrument aperture at the sun for solar radiometric calibrations. (b) Glint observations are made at latitudes on the sunlight hemisphere where the solar zenith angle is less than 75°. (c) For Target observations, the spacecraft points the instrument at a stationary surface target as it flies over. A small-amplitude sinusoidal oscillation in the pitch axis is superimposed on the nominal pointing to scan the spectrometer slit across the Target.*

will be less than 24 seconds, such that the slit is scanned over the target >20 times in a 9-minute Target observation.

For Nadir and Glint observations, the ACS is required to point the instrument's field of view to within 0.25° of its intended target (3 km at nadir). For Target observations, a pointing accuracy of 0.3° is required. OCO-2 will switch from Nadir to Glint observations on alternate 16-day global ground-track repeat cycles so that the entire OCO-2 ground track is mapped in each mode every 32 days. Comparisons between Nadir and Glint observations will provide opportunities to identify and correct for biases introduced by the viewing geometry. Target observation will be acquired over an OCO-2 validation site roughly once each day.

The same rate of data sampling is used for Nadir, Glint, and Target observations. The instrument collects 8 adjacent, spatially resolved samples every 0.333 seconds (24 samples per second). At this data collection rate, the Observatory collects ~400 soundings per degree of latitude as it travels from pole to pole, or ~14 million soundings over the sunlit hemisphere every 16 day ground repeat cycle. Clouds, aerosols, and other factors will reduce the number of soundings available for X_{CO2} retrievals, but the small sounding footprint ensures that some data will be sufficiently cloud free on regional scales at monthly intervals.

Nadir observations will be collected at all locations where the surface solar zenith angle is less than 85°. This mode provides the highest spatial resolution and is expected to return more usable soundings in regions that are partially cloudy or have significant surface topography. However, Nadir observations are expected to have limited SNRs over dark ocean- or ice-covered surfaces at high latitudes. Glint observations are expected to provide much greater SNR over these surfaces. Glint soundings will be collected at all latitudes where the surface solar zenith angle is less than 75°. Target observations will be conducted over OCO-2 validation sites that are within 61° of the local spacecraft nadir along the orbit track and spacecraft viewing angles between 30° west of the ground track and 5° east of the ground track. When the target is near the ground track, a single pass can last for up to 9 minutes, providing 12000 to 24000 soundings in the vicinity of the target. This large number of soundings reduces the impact of random errors and provides opportunities to identify spatial variability in the X_{CO2} field near the target.

While the sunlight incident at the top of the Earth's atmosphere is not polarized, both reflection from the surface and scattering by the atmosphere can affect the polarization of the radiation field measured by the OCO-2 instrument. These processes act primarily to reduce the intensity of the radiation that is polarized in the direction parallel to the Principal Plane. Polarization has a much smaller effect on the intensity polarized in the direction perpendicular to the Principal Plane. As noted above, the OCO-2 instrument is only sensitive to light polarized in the direction parallel to the orientation of the long axis of the spectrometer slits. The Nadir and Glint observing strategies have therefore been designed such that the long axis of the spectrometer slits (which are roughly parallel to the Observatory y-axis) remains oriented perpendicular to Principal Plane to maximize signal and minimize polarization errors (Figure 2-4a). As the Observatory ascends over the southern terminator, its x-axis is pointed north-northwest along the orbit track and the spectrometer slits are oriented almost perpendicular to the orbit track (Figure 2-4b). In this orientation, the instrument collects data in a conventional push-broom fashion, where the footprint is determined by the cross-track instantaneous field of view (0.1°) and the integration time (0.333 seconds). For Nadir observations, this yields 4 to 8 cross-track footprints along the spectrometer slit with dimensions of 1.29 km by 2.25 km.

As the Observatory proceeds northward along its orbit, it rotates counterclockwise about its z-axis, such that the x-axis points westward, and the long axis of the spectrometer slits are aligned with the track just north of the sub-solar latitude. At this point, each spatially resolved surface footprint is determined by the projected width of the slit (<0.03°) and the exposure time. For Nadir observations at the sub-solar latitude, each of the footprints is ~0.4 km by 2.25 km and there is spatial overlap between footprints acquired in successive exposures by the spatial elements along the slit. The Principal Plane azimuth rotation continues as the Observatory approaches the northern terminator, where the x-axis is pointing southwest, along the orbit track, and the spectrometer slit is once again almost perpendicular to the orbit track.

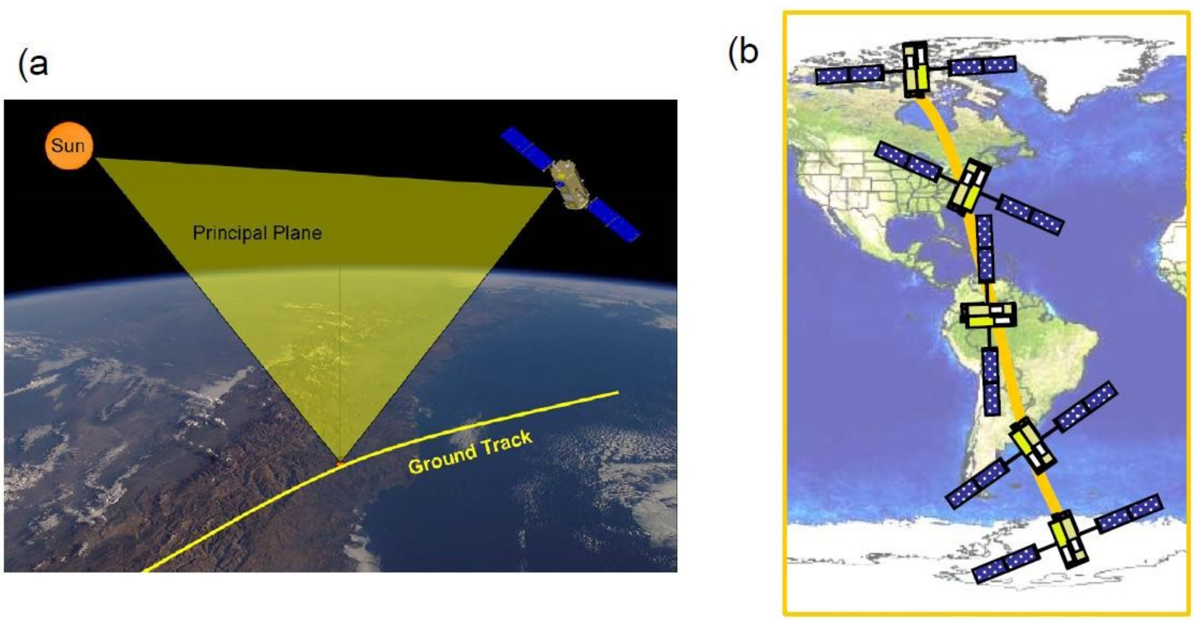

Figure 2-4. *(a) The Principal Plane is defined with respect to the sun, surface footprint and spacecraft. (b) The spacecraft azimuth changes during the orbit to maintain the alignment of the spectrometer slits (which are roughly parallel with the axis of the solar panels) perpendicular to the Principal Plane [Crisp et al 2007].*

2.2.2 Data product delivery

Science and housekeeping data are transmitted to a NASA Near Earth Network station in Alaska once each day. The data are then transferred to the Earth Science Mission Operations (ESMO) center at the NASA Goddard Space Flight Center (GSFC) where the raw telemetry is converted to time-ordered raw radiance spectra (Level 0 Products). This product is then delivered to the OCO-2 Science Data Operations System (SDOS) at the NASA Jet Propulsion Laboratory, where full orbits are first processed to yield radiometrically calibrated, geolocated spectral radiances within the O_2 and CO_2 bands (Level 1 Products). The bore-sighted spectra for each coincident CO_2/O_2 sounding are then processed to estimate the column averaged CO_2 dry air mole fraction, X_{CO2} (Level 2 Products). Other Level 2 data products to be retrieved from each sounding include the surface pressure, surface-weighted estimates of the column-averaged water vapor and atmospheric temperature, the vertical distribution and optical depth of optically-thin clouds and aerosols, the CO_2 column averaging kernels, and a number of diagnostic products.

2.3 OCO-2 Algorithm

2.3.1 Level 2 algorithm overview

The FP X_{CO2} retrieval algorithm was derived from the algorithm developed for the OCO. It was further refined, in the time between the OCO and OCO-2 launches, by use in producing the Atmospheric CO_2 Observations from Space (ACOS) data product. The algorithm is a Rodgers (2000)-type optimal estimation approach and has been described fully in O'Dell et al. (2011). The retrieval algorithm consists of a forward model, an inverse method, and an error analysis step. The overall flow for the retrieval process is shown in Figure 2-5.

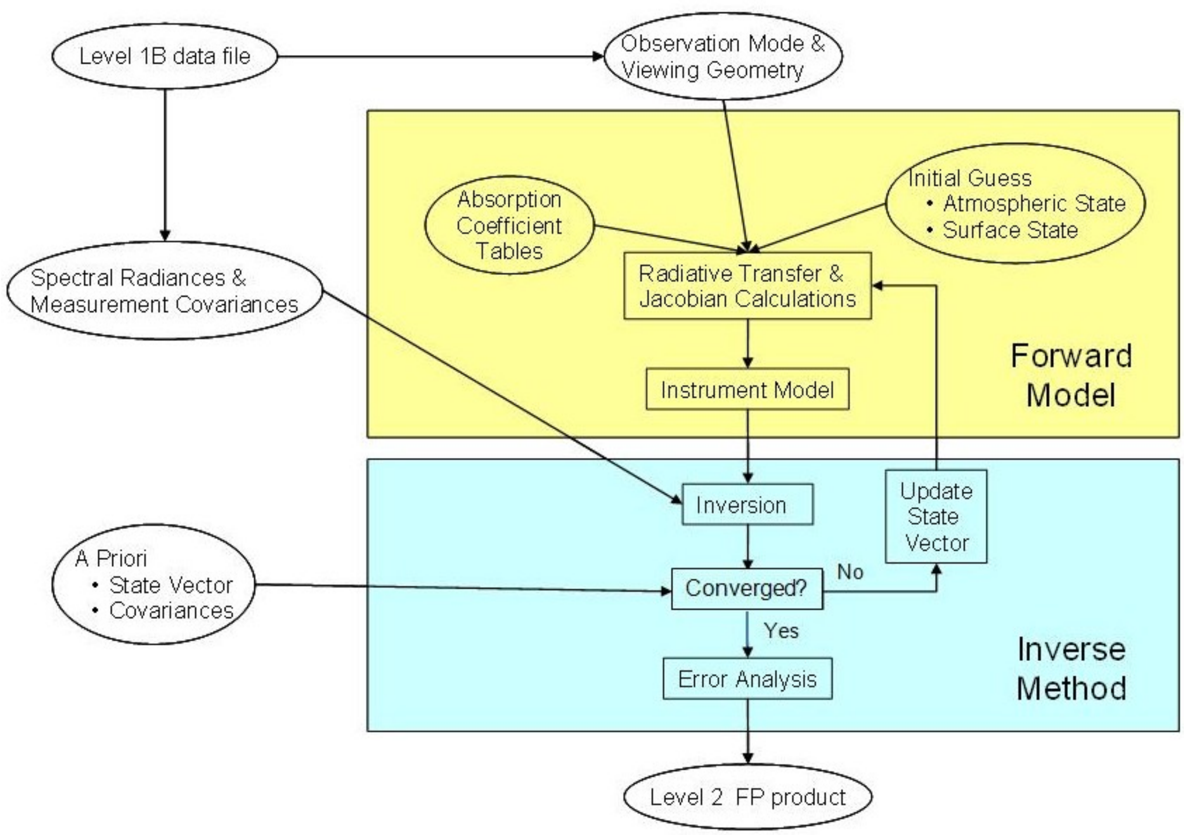

Figure 2-5. *Level 2 full physics retrieval flow.*

A forward radiative transfer model is used to generate synthetic spectra within the molecular O_2 A-band at 0.76 microns and the weak and strong CO_2 bands centered near 1.61 and 2.06 microns, respectively. These synthetic spectra are then convolved with the OCO-2 instrument line shape and compared to the observed spectra in each of these spectral regions. An inverse model then modifies the assumed atmospheric state to improve the fit to the measured spectra, and the process is repeated until the convergence criteria are met. The forward radiative transfer model contains components simulating the solar spectrum, absorption by CO_2, O_2, H_2O, and other gases, scattering and absorption by clouds and aerosols, reflectance of the surface. Input to the forward model consists of meteorological conditions, surface properties, characteristics of the instrument, etc. The forward model returns simulated radiance spectra and the partial derivatives of those radiances with respect to properties of the atmospheric and surface state, also called Jacobians.

The residuals between the simulated and measured spectra are minimized by changing the properties of the atmospheric and surface state via the inverse method. This inversion uses the Jacobians to estimate the state changes needed to minimize the differences between the observed and simulated spectra.

Once the atmospheric state yielding the best match to the observed spectrum has been found, the algorithm then determines X_{CO2}, errors in X_{CO2} from different sources (such as vertical smoothing, measurement noise, etc.), and the X_{CO2} column averaging kernel. This is necessary because X_{CO2} is not itself an element of the state vector. Rather, it is determined from the profile of CO_2, which is part of the state vector. It is formally given by the total number of CO_2 molecules in the column divided by the total number of dry air molecules in the column. This step is labeled "Error Analysis" in Figure 2-5.

3 Overview of Data Products

All OCO-2 product files are in Hierarchical Data Format-5 (HDF), developed at the National Center for Supercomputing Applications (http://www.hdfgroup.org). This format facilitates the creation of logical data structures.

3.1 File Naming Conventions

The OCO-2 L2Std files follow this convention:

oco2_L2Std[Mode]_[Orbit][ModeCounter]_[AcquisitionDate]_[ShortBuildId]_[ProductionDateTime][Source].h5

The OCO-2 L2Dia files follow this convention:

oco2_L2Dia[Mode]_[Orbit][ModeCounter] _[AcquisitionDate]_[ShortBuildId]_[ProductionDateTime][Source].h5

The OCO-2 L2IDP files follow this convention:

oco2_L2IDP[Mode]_[Orbit][ModeCounter]_[AcquisitionDate]_[ShortBuildId]_[ProductionDateTime][Source].h5

The OCO-2 L1bSc files follow this convention:

oco2_L1bSc[Mode]_[Orbit][ModeCounter]_[AcquisitionDate]_[ShortBuildId]_[ProductionDateTime][Source].h5

For all files, the fields are defined as below.

- [Mode] is the acquisition mode as a two character string:
  - GL—Sample Glint
  - ND—Sample Nadir
  - TG—Sample Target
  - DS—Sample Dark Calibration
  - LS—Sample Lamp Calibration
  - SS—Sample Solar Calibration
  - BS—Sample Limb Calibration
  - NP—Single-Pixel Nadir
  - GP—Single-Pixel Glint
  - TP—Single-Pixel Target
  - DP—Single-Pixel Dark Calibration
  - LP—Single-Pixel Lamp Calibration
  - SP—Single-Pixel Solar Calibration
  - BP—Single-Pixel Limb Calibration
  - XS—Sample Transition
  - XP—Single-Pixel Transition
  - MS—Sample Lunar Calibration
  - MP—Single-Pixel Lunar Calibration
  - SB—Stand-by

- [Orbit] is the five-digit orbit number
- [ModeCounter] is a letter (a, b, c, d) denoting the times an acquisition mode occurred in an orbit. If a mode occurs only once, ModeCounter is set to "a"
- [AcquisitionDate] is the UTC date (yymmdd) the data were acquired
- [ShortBuildId] identifies the L1b build version used (Bn.m.uu) where n is the major version, m is subversion number, and the uu is the incremental/patch number
- [ProductionDateTime] is the date and time the file was produced (yymmddhhmmss)
- [Source] (if present) identifies production sources different from the standard operations pipeline. This field will be missing from normal pipeline data.

3.1.1 File Format and Structure

The OCO-2 product files contain data structures indexed by sounding (1 to N soundings/file) and are associated by the sounding_id variable in all products.

Variables are combined into groups by type (e.g., SoundingGeometry). Within each type, a variable has one or more values per sounding. Variables may be single-valued (e.g., *sounding_altitude*) or multi-valued (e.g., *co2_profile*).

The metadata of each variable describes the variable's attributes, such as dimensions, data representation, and units.

Note that many variables in the L2 products use *_fph* to denote full physics algorithm, *_idp* for IMAP-DOAS, and *_abp* for the O_2 A-band preprocessor. For example, surface pressure is calculated in each of the algorithms, so it is reported with a tag on the variable to differentiate them.

3.1.2 Data Definition

The OCO-2 data products contain many variables with a variety of dimensions. The following list describes only the most important of the dimensions. Dimensions and data shapes are fully described within the hdf files.

- Retrieval—the number of retrievals reported (those soundings for which retrievals converged or were converging when the maximum number of iterations was reached)
- Band—the three bands of OCO-2 are O_2 A, weak CO_2, and strong CO_2
- Footprint—the eight footprints across the swatch are identified as 1 to 8
- Sample—the spectral element. Each band has 1016 spectral elements, although some are masked out in the L2 retrieval
- Sounding—one set of measurements (one footprint across three bands) that is the primary unit for retrievals

3.1.3 Global Attributes

In addition to variables and arrays of variables, global metadata is stored in the files. The granule-level metadata is described in Table 8-1. ECHO metadata and other metadata related to HDF version and production location can be found in the hdf file but are not discussed here.

3.2 File Content

The relationship of the overall data flow and retrieval processes to the product files is illustrated in Figure 3-1.

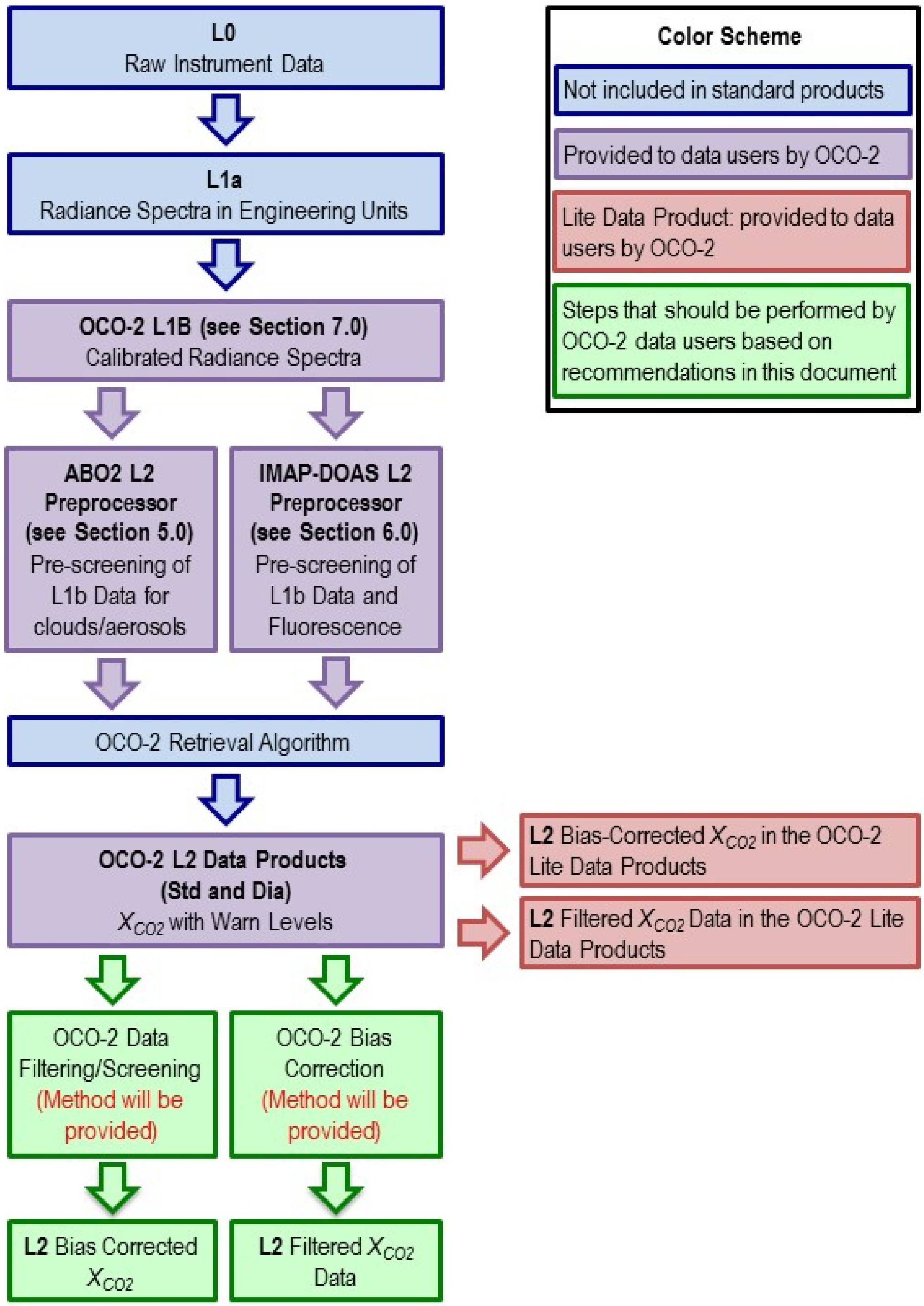

Figure 3-1. *An illustration of the flow through the OCO-2 data retrieval process, highlighting some of the tasks that could be performed by the data user.*

3.2.1 L2Std

Geolocated retrieved CO_2 column averaged dry air mole fraction - physical model. This file contains the retrieved values for the state vector as well as geolocation information. A number of fields from the Iterative Maximum a Posteriori Differential Optical Absorption Spectroscopy (IMAP-DOAS) and O_2-A Band (ABO2) cloud preprocessors are brought forward to the L2 products.

3.2.2 L2Dia

This file has the data that is in the L2Std, as well as averaging kernels, a priori covariance matrices, and posterior covariance matrices.

3.2.3 L2IDP

There are the results from the IMAP-DOAS process. The IMAP-DOAS process is used in cloud screening, but also provides solar induced fluorescence (SIF) measurements for a much larger set of data then contained in L2.

3.2.4 L1bSc

These are the radiance spectra that are the input to the L2 retrievals.

4 OCO-2 L2 Data Products

4.1 Data Description and User Alerts

The data products include both Standard files and Diagnostic files. While much of the data in the two file types are the same, Section 4.3 describes a few of the fields that are uniquely in the L2Dia file.

Note that warn levels are now included in the data. The point of this field is to indicate the likely quality of the data. They are a value that ranges from 0 to 19, which provides the user more flexibility to select data that they want to use in analysis. See more details in Section 4.2.1.2.

A subset of fields is discussed here (Figure 4-1)Figure 4-1. The later section contains full data field tables.

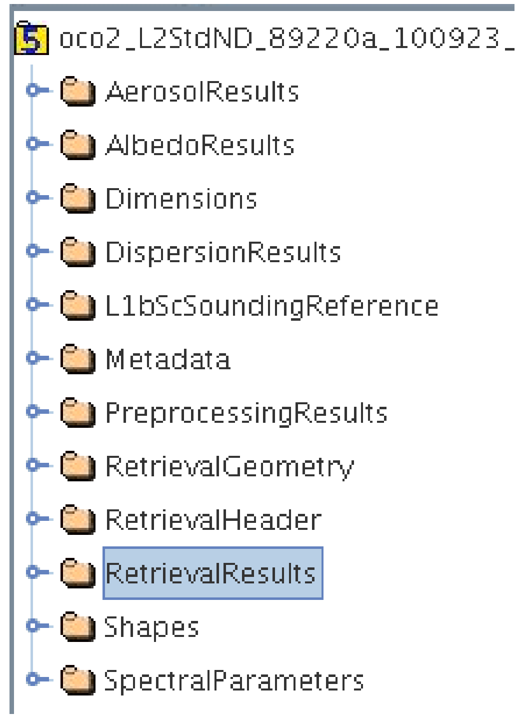

Figure 3-1. *Folders contained in the L2Std product.*

4.2 Key Data Fields for Standard and Diagnostic Files

The data and h5 folders that are included in the standard files are also included in the diagnostic files.

4.2.1 RetrievalHeader

Figure 4-2 shows the variables contained in the RetrievalHeader folder.

4.2.1.1 Sounding_ID

This is the label for the sounding that will let you link the data across all of the product files. The *sounding_id* is the primary identification number for soundings across all OCO-2 products. The *sounding_id* is a composite of the time the data was acquired and the footprint number (f) in the form yyyymmddhhmmsssf.

4.2.1.2 Warn_level

The warn levels are provided to assist the user in selecting data. Warn levels range from 0 to 19. As discussed in Mandrake et al (2013), the low values are higher quality data. For each data release, we will include information about the standard deviation of the XCO2 compared to validation data and how that changes with warn levels. We will also provide some recommendations about warn levels – how much data you will get by screening different ways, etc. For the sample data, the warn levels have an appropriate range of values, but otherwise are not meaningful.

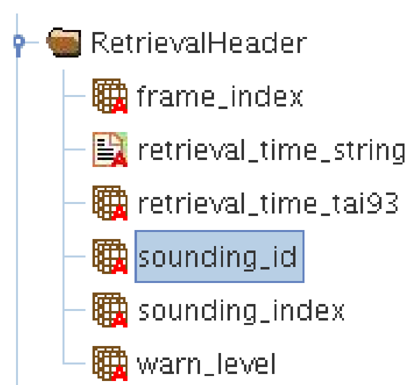

Figure 4-2. *Variables in the RetrievalHeader folder.*

4.2.1.3 Retrieval_time_string

This is the time as a string. An example of this field is 2010-09-23T18:36:04.334Z.

4.2.1.4 Retrieval_time_tai93

This field is the time of the measurement, in seconds since Jan. 1. 1993. The TAI time corresponding to the example above is 559.420571E6.

4.2.2 RetrievalGeometry

Figure 4-3 shows the variables contained in the RetrievalGeometry folder.

4.2.2.1 Retrieval_latitude

This is the latitude of the sounding. These values range from -90 to 90.

4.2.2.2 Retrieval_longitude

This is the longitude. We use the convention of -180 to 180 for longitude, with fractional degrees.

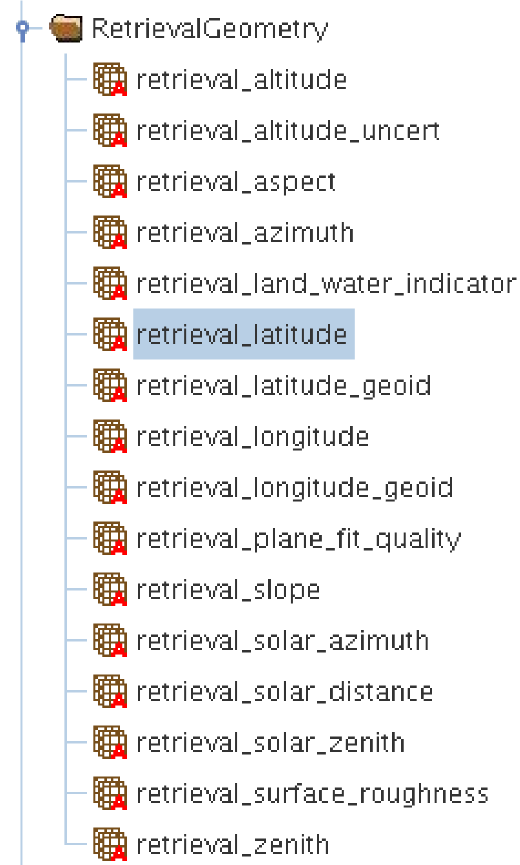

Figure 4-3. *Variables in the RetrievalGeometry folder.*

4.2.3 RetrievalResults

The retrieval results folder contains all of the elements of the state vector in the L2 Full Physics retrieval.

4.2.3.1 XCO2

This variable expresses the column-averaged CO_2 dry air mole fraction for a sounding. Those soundings that did not converge will not be present. These values have units of mol/mol. This can easily be converted to ppm by multiplying by 10^6.

4.2.3.2 XCO2_uncert

This is an estimate of the uncertainty on the reported X_{CO2}. These values have units of mol/mol.

4.2.3.3 XCO2_apriori

This is the initial guess for the XCO2, in mols/mol.

4.2.3.4 XCO2_avg_kernel

This is the averaging kernel. Note that the normalized Averaging Kernel (RetrievalResults/xco2_avg_kernel_norm) for a given pressure level is equal to the non-normalized value (retrieval-results/xco2_avg_kernel) divided by the pressure weighting function at that level. Note that levels are "layer boundaries" and have no thickness. See Appendix A of O'Dell et al. (2012) for details on how these quantities are defined.

4.2.3.5 XCO2_avg_kernel_norm

See description in Section 4.2.3.4.

4.3 Key Data Fields Diagnostic Files

The Diagnostic files provide additional information that will be of use to scientists performing data assimilation.

4.3.1 RetrievalResults

The RetrievalResults folder for Diagnostic files has variables not included in the RetrievalResults folder for Standard products. These variables include the covariance matrices (a priori and a posteriori) and the averaging kernel matrix, as well as estimates of uncertainty from interference (interference smoothing uncertainty and Xco2_correlation_interf).

The measured and model radiances are also in this expanded file (along with a wavelength grid). The measured radiance matches what is found in the L1b file, and the modeled is the radiance calculated in the L2 Full Physics at the last retrieval step.

4.3.1.1 Averaging_kernel_matrix

This matrix is the averaging kernel for all state vector elements.

4.3.1.2 Aposteriori_covariance_matrix

This field is the a posteriori covariance matrix for all elements of the state vector. Note that there is also a separate variable for the co2_profile_covariance_matrix.

4.3.2 RetrievedStateVector

The RetrievedStateVector folder is unique to the Diagnostic files. It contains all of the state vector variables as a vector of the variables. All of these have been separately reported in the RetrievalResults folder as scalars.

5 ABO2 Preprocessor

The O_2-A Band cloud screening algorithm was developed at Colorado State University (CSU) under the ACOS program. It employs a fast Bayesian retrieval to estimate surface pressure and surface albedo, assuming clear-sky conditions with only molecular Rayleigh scattering, from high-resolution spectra of the O_2A-band near 765 nm. The estimated surface pressure, surface albedo and the chi-squared goodness-of-fit statistic are used to flag scenes as cloudy, clear, or indeterminate [Taylor, *TGRS*, 2011].

The basic method is that, absent clouds or aerosols, the surface pressure of a clear scene can be determined to within 2-5 hPa accuracy using the O_2A-band spectrum of reflected sunlight. This is because of the strong oxygen absorption features that are present in this band. When surface pressure is higher, the absorption features are deeper for a given observation geometry. When clouds or aerosols are present, they change the path lengths for most photons, either via shortening or lengthening, such that the retrieved surface pressure can be very different from the expected value based on a meteorological forecast, and also the O_2A spectrum itself cannot be well-fit with a clear-sky assumption.

5.1 Prescreening of OCO-2 Soundings for Cloud and Aerosol

The primary objective of the ABO2 algorithm is to remove from the OCO-2 operational processing stream the soundings that are deemed too contaminated by clouds and /or aerosol for reliable X_{CO2} retrieval in the computationally expensive L2 algorithm. The key screening parameters are interpreted internally via thresholds to provide a simple cloud flag. Furthermore, the key screening parameters can be used as inputs to a genetic algorithm used for individual sounding selection [Mandrake, *AMT*, 2013]. Details of the technique will be described in the ATBD as well as an upcoming publication. As this is used as a preprocessor, the primary objective is to flag as cloudy scenes that are obviously cloudy or aerosol-contaminated. Therefore, thresholds are set rather loosely so scenes that have some cloud or aerosol contamination will sometimes pass the filter. This is by design. These scenes are sometimes useful for science, or may be rejected by additional pre- or post-processor flags.

5.2 Key Science Data Fields

5.2.1 PreprocessingResults

The ABO2 data fields can be found in the standard L2 product (L2Std) PreprocessingResults folder, labeled with the *_abp* data field designation.

Figure 5-1 shows a screenshot of an example OCO-2 L2Sc product file as viewed by HDFView. The ABO2 data fields within the PreprocessingResults folder are highlighted and will be described in detail below.

5.2.1.1 surface_pressure_delta_abp

The delta surface pressure is calculated as ECMWF standard – retrieved pressure – the offset:

surface_pressure_apriori_abp - surface_pressure_abp - surface_pressure_offset_abp

in SI units (Pascals). This is the primary screening criteria for ABO2, and represents the difference between the retrieved and meteorologically estimated surface pressure in the target field-of-view. Because the algorithm uses imperfect spectroscopy to retrieve surface pressure, there is a path length dependent offset term that is determined by analyzing clear scenes and subtracted from the retrieved value. This yields an unbiased estimate of the retrieved surface pressure. A threshold value can be set independently for nadir, glint, and target viewing modes. At the time of this writing, scenes with a difference greater than 25 hPa (2500 Pa) are flagged as cloudy, for all OCO-2 viewing modes.

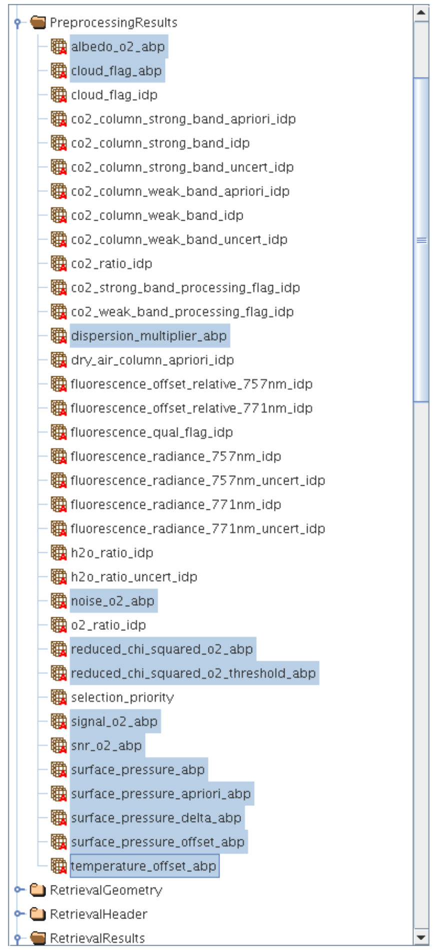

Figure 5-1. *Screenshot of an HDFView look at the ABO2 preprocessor file.*

5.2.1.2 albedo_o2_abp

The retrieved surface albedo at 0.755 μm and 0.785 μm. Note that the retrieval assumes a perfect Lambertian surface albedo that varies linearly with wavelength. This assumption is currently made for all viewing modes; in glint mode, this means that the retrieved surface albedo can sometimes exceed unity. Over a dark ocean surface, the retrieved surface albedo is a good way to tell the presence of cloud: if the retrieved surface albedo is too large, a cloud or reflective aerosol layer is likely present. This test is not useful in glint mode or over land, and therefore is generally not used for OCO-2 prescreening.

5.2.1.3 reduced_chi_squared_o2_abp

The reduced χ^2 value of the spectral fit of the fast retrieval. Values greater than a threshold value are also indicative of clouds or aerosols present; as they are not accounted for in the retrieval, spectra containing them cannot be well fit. The threshold χ^2 value is a parameterized function of SNR, as there are persistent spectral features (due to imperfect spectroscopy) that scale with the signal level.

5.2.1.4 dispersion_multiplier_abp

This parameter is also fit in the retrieval, and accounts for wavelength shifts in the spectra due primarily to the earth-instrument Doppler shift. This is typically a maximum of ±7 km/sec, which corresponds roughly to shifts of ±0.3 cm^{-1} or ±0.018 nm (at maximum). This Doppler shift is easily fit for in the retrieval. Because the Doppler shift is formally a wavelength scaling, the ABO2 algorithm fits it as a scaling rather than a simple shift, though because of the narrowness of the fitted spectral region, it amounts to roughly the same thing.

5.2.1.5 noise_o2_abp

The determined radiance noise value in the continuum based on the preflight calibration.

5.2.1.6 reduced_chi_squared_o2_threshold_abp

The threshold reduced χ^2 as described above. The logarithm of the reduced χ^2 in clear scenes is assumed to be a piecewise linear function of SNR. This relationship is determined from clear scenes separately for nadir and glint modes.

5.2.1.7 signal_o2_abp

The determined radiance mean value in the continuum.

5.2.1.8 snr_o2_abp

The ratio of signal_o2_abp to noise_o2_abp.

5.2.1.9 surface_presssure_abp

The retrieved value of surface pressure by the ABO2 algorithm, in units of Pa.

5.2.1.10 surface_presssure_apriori_abp

The prior estimate of surface pressure in the target field-of-view, as determined from short-term ECMWF meteorological forecasts, and adjusted based on the mean elevation in the target field of view.

5.2.1.11 surface_presssure_offset_abp

The offset of the retrieved surface pressure for clear sky scenes. At the time of this writing, these are all set to zero as there is no data for which to determine the actual OCO-2 offsets. After some inflight data are obtained, the offset will be determined as a piecewise linear function of solar zenith angle, separately for nadir and glint modes.

5.2.1.12 temperature_offset_abp

The retrieved temperature offset, a simple additive offset to the meteorological estimate of the temperature profile. At the time of this writing this parameter is included in the state vector, but may be taken out to increase processing speed.

5.2.1.13 cloud_flag_abp

The result of the surface pressure, χ^2, and albedo tests to determine if a scene is likely cloudy (or heavily aerosol-laden). If a single test is failed, the cloud flag is set equal to 1, indicating the likely presence of cloud and/or aerosol. If all threshold checks are successfully passed, then the cloud flag is assigned a value of 0, indicating a sufficiently clear-sky scene. The cloud flag can also be set to 2 for "undetermined" cases. These are chiefly caused by solar zenith angle out of bounds or when viewing water surfaces in nadir observation mode (insufficient SNR).

6 IMAP-DOAS Preprocessor

The Iterative Maximum a Posteriori Differential Optical Absorption Spectroscopy preprocessor is a non-scattering fast retrieval algorithm for optically thick absorbers [Frankenberg et al, *ACP*, 2005]. The preprocessor now serves two purposes for OCO-2: (1) retrieve vertical columns of trace gases for advanced cloud and aerosol screening and (2) retrieve solar induced chlorophyll fluorescence (SIF) using an algorithm described in Frankenberg et al, *GRL* (2011).

6.1 Advanced Cloud and Aerosol Screening

For OCO-2, it is being used to derive vertical column densities of H_2O and CO_2 independently in both CO_2 bands (1.6 and 2.0µm) under the assumption of a non-scattering atmosphere. Given that Rayleigh scattering is very low in the near-infrared, this assumption holds true if neither aerosols nor clouds are present. In this case, both bands should yield an accurate and consistent result, i.e. the ratio of retrieved quantities in both CO_2 bands is approaching unity.

We found that the ratio of both CO_2 and H_2O vertical column densities is deviating significantly from unity in the presence of aerosols and clouds (e.g., Mandrake et al, *AMT* 2013). The details of this technique will be described in the Algorithm Theoretical Basis Document (ATBD) as well as an upcoming publication.

6.2 Retrievals of Solar-Induced Chlorophyll Fluorescence

The possibility of retrieving solar induced chlorophyll fluorescence from high-resolution spectra in the vicinity of the O_2 A-band have been shown in Frankenberg et al 2011 and Joiner et al 2011. Here, we apply the retrieval algorithm described in Frankenberg et al 2011, embedded in the IMAP-DOAS retrieval code. It is important to note that this is the dedicated fluorescence retrieval, as opposed to the fluorescence retrieval within the full-physics L2 code. Interference with scattering properties in this retrieval is thus minimized (e.g., Frankenberg et al, *AMT* 2012).

6.3 Key Science Data Fields

Note: There are a few generic differences between the IMAP-DOAS data and the main L2 files based on the full-physics X_{CO2} retrieval. First of all, all OCO-2 L1bSc data with a solar zenith angle smaller than 80 degrees and a valid quality flag are processed through the preprocessor. The other main difference is the structure of the IMAP L2 data fields as they are arranged in 2 dimensional data fields, with one dimension being time of readout (variable per orbit) and the other one representing each of the 8 OCO-2 footprints independently (i.e., the dimensions are n x 8, with n denoting the number of total focal plane array readouts per orbit).

Figure 6-1 shows a screenshot of the general structural overview of an IMAP-DOAS preprocessor L2 file using HDFView. The most important data folders are expanded and will be described in detail below.

6.3.1 SoundingGeometry

The SoundingGeometry folder includes all relevant location information for each footprint as well as information on viewing geometries as well as topography. It follows exactly the same definitions as the official full-physics L2 files and the reader is referred to the more detailed explanation in the respective document for the official X_{CO2} L2 file.

6.3.2 DOASCO2

The DOASCO2 folder contains all relevant fields from the IMAP retrievals of CO_2 columns in both OCO-2 CO_2 bands. Definitions are as follows:

6.3.2.1 DOASCO2/co2_column_strong_band_apriori_idp

A priori vertical column density of CO_2 in the strong (2.0µm) band in molecules/m^2. Note that a constant volume mixing ratio (VMR) is assumed across the globe in this IMAP-DOAS version and that the a priori thus mostly depends on surface pressure.

6.3.2.2 DOASCO2/co2_column_weak_band_apriori_idp

See above but for weak band (both are identical).

6.3.2.3 DOASCO2/co2_column_strong_band_idp

Retrieved vertical column density of CO_2 in the strong (2.0µm) band in molecules/m2. Note that this is different from the full-physics (FP) L2 CO_2 vertical column as this variable here is only retrieved in one band and ignores scattering.

6.3.2.4 DOASCO2/co2_column_weak_band_idp

See above but for the weak band.

6.3.2.5 DOASCO2/co2_column_strong_band_uncert_idp

Uncertainty in retrieved vertical column density of CO_2 in the strong (2.0µm) band in molecules/m^2.

6.3.2.6 DOASCO2/co2_column_weak_band_uncert_idp

See above but for weak band.

6.3.2.7 DOASCO2/co2_column_strong_band_processing_flag_idp

Processing flag for the strong CO_2 band retrieval. Every sounding will have a value: 0 (successfully processed), 1 (not converged) or 2 (not processed).

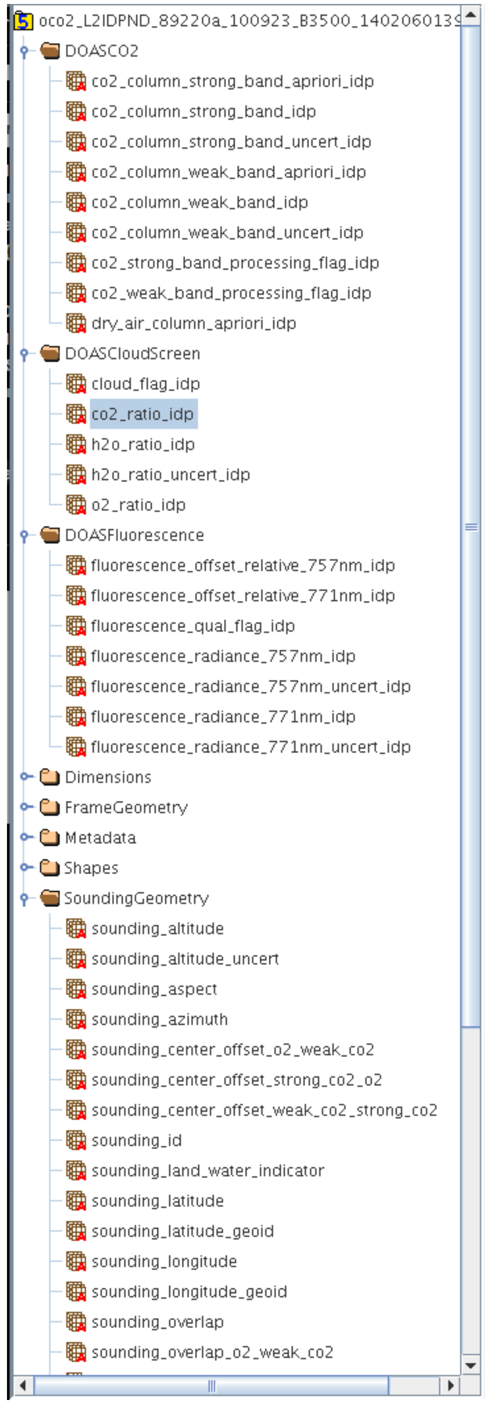

Figure 6-1. *Screenshot of an HDFView look at the IMAP-DOAS preprocessor file.*

6.3.2.8 DOASCO2/co2_column_weak_band_processing_flag_idp

See above but for weak band.

6.3.2.9 DOASCO2/dry_air_column_apriori_idp

A priori total dry air column (molec/m^2) of the respective sounding (based purely on ECMWF input data and topography).

6.3.3 DOASCloudScreen

In this folder, all relevant variables for advanced cloud-screening are stored.

6.3.3.1 DOASCloudScreen/cloud_flag_idp

Tentative cloud flag based purely used on the IMAP preprocessor. Values can be:

- -2=unusable (not processed)
- -1=not all retrievals converged
- 0=clearly cloudy
- 1=probably cloudy
- 2=probably clear
- 3=very clear

The values are based on simple threshold criteria and based on the ratios defined in the same folder. This flag is NOT being used as input for FP L2 sounding selection and is also not being validated.

6.3.3.2 DOASCloudScreen/co2_ratio_idp

Ratio of retrieved CO_2 vertical column density (VCD) in weak and strong band ($VCD_{1.6\mu m}/VCD_{2.0\mu m}$).

6.3.3.3 DOASCloudScreen/h2o_ratio_idp

Ratio of retrieved H_2O vertical columns in weak and strong band ($VCD_{1.6\mu m}/VCD_{2.0\mu m}$).

6.3.3.4 DOASCloudScreen/h2o_ratio_uncert_idp

1-sigma uncertainty of the ratio of retrieved H_2O vertical columns in weak and strong band ($VCD_{1.6\mu m}/VCD_{2.0\mu m}$).

6.3.3.5 DOASCloudScreen/o2_ratio_idp

Ratio of retrieved vs. expected O_2 column density based on the fluorescence 771 nm retrieval window, which includes some weak O_2 line used to derive an O_2 column. Values substantially lower than 1 indicated the presence of clouds.

6.3.4 DOASFluorescence

In this folder, all relevant variables for the fluorescence retrieval in the 757 nm and 771 nm window are saved. From experience with Greenhouse Gases Observing Satellite (GOSAT) and also looking at the general shape of the fluorescence emission spectrum, SIF at 771 nm is about a factor 1.7 lower than at 757 nm.

Please note that the primary retrieval fits the relative contribution of an additive offset (such as from fluorescence) in both retrieval windows. Over non-fluorescing targets, the average retrieval of this offset may deviate from 0 because of small uncertainties in instrument line-shape as well as the solar line-list. In relative units, this "bias" is expected to be constant across the globe but a constant bias in the relative contribution to of fluorescence will translate into a signal-dependent bias in the absolute value (in radiance units). We will characterize this bias as soon as possible but users, esp. in the early mission phase, should be aware of that caveat and provide feedback of observed inconsistencies to the developer team.

6.3.4.1 DOASFluorescence/fluorescence_offset_relative_757nm_idp

Fraction of continuum level radiance explained by an additive offset term in the 757 nm spectral window (unitless). In the absence of instrumental errors, this will be only caused by fluorescence. Rotational Raman scattering should be negligible over typical vegetated surface and moderate solar zenith angles (<65 degrees).

6.3.4.2 DOASFluorescence/fluorescence_offset_relative_771nm_idp

Same as above but for the 771 nm window.

6.3.4.3 DOASFluorescence/fluorescence_radiance_757nm_idp

Fluorescence term expressed in absolute radiance units (ph/s/m^2/sr/µm) in the 757 nm fit window. This is a derived quantity based on the fit of the relative contribution multiplied with the overall continuum level radiance.

6.3.4.4 DOASFluorescence/fluorescence_radiance_771nm_idp

Same as above but for the 771 nm window.

6.3.4.5 DOASFluorescence/fluorescence_radiance_757nm_uncert_idp

Estimated uncertainty (1-sigma) of the fluorescence term expressed in absolute radiance units (ph/s/m^2/sr/µm) in the 757 nm fit window.

6.3.4.6 DOASFluorescence/fluorescence_radiance_771nm_uncert_idp

Same as above but for the 771 nm window.

6.3.4.7 DOASFluorescence/fluorescence_qual_flag_idp

Quality flag for the fluorescence retrieval (0=ok, 1=didn't pass quality filter or unprocessed).

7 OCO-2 L1bSc Data Products

The OCO-2 L1bSc products contain the calibrated spectra that are used as input for the X_{CO2} products (Figure 7-1). These files contain all acquired data, and consequently are quite voluminous. Additionally, complete geolocation information for all of the footprints as well as some instrument and calibration information can be found the L1bSc products.

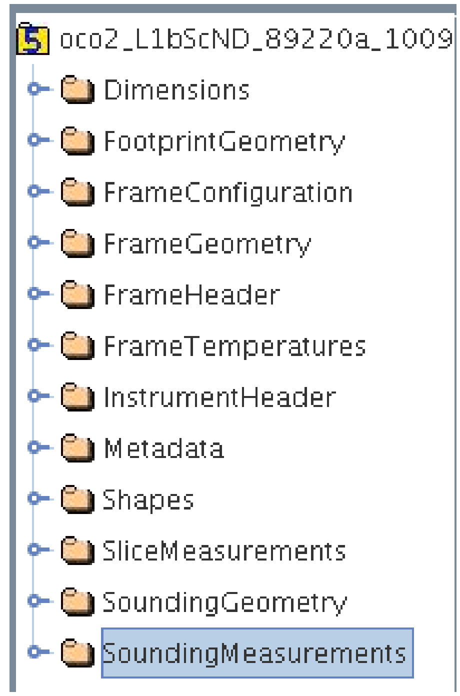

Figure 6-1. *Folders in the L1bSc product.*

7.1 Key Data Fields

7.1.1 FootprintGeometry

7.1.1.1 *Footprint_latitude*

Note that these are for each band, unlike the *retrieval_latitude*, that is for a sounding.

7.1.1.2 *Footprint_longitude*

Note that these are also for each band, using a -180 to 180 convention.

7.1.2 SoundingMeasurements

7.1.2.1 *Radiance_o2*

This is reported for each of the eight footprints. Note that a field of the SNR coefficients will indicate if a sample should be used or not. (Early sample data will not have this implemented correctly.)

7.1.2.2 *Radiance_strong_co2*

The radiance array for the strong CO_2 band.

7.1.2.3 *Radiance_weak_co2*

Radiance array for the weak CO_2 band.

7.1.3 InstrumentHeader

7.1.3.1 *dispersion_coef_samp*

Coefficients that express the relationship between the spectral element index and its associated wavelength. Note that this grid does not account for the Doppler correction or dispersion adjustments that are applied in L2. The coefficients are used as follows:

$$\lambda = \sum_{i=0}^{5} c_i \ . \ \text{column}^i$$

where column refers to the column number in the L1bSc files (0 to 1016).

An example calculation of the wavelength grid is:

$$\begin{aligned}\lambda &= 0.757633 + 1.75265 \text{ x } 10^{-5} \cdot \text{column}^1 \\ &- 2.91788 \text{ x } 10^{-9} \cdot \text{column}^2 + 3.29430 \text{ x } 10^{-13} \cdot \text{column}^3 \\ &- 2.72386 \text{ x } 10^{-16} \cdot \text{column}^4 + 7.66707 \text{ x } 10^{-20} \cdot \text{column}^5\end{aligned}$$

7.1.3.2 SNR_coef

The SNR coefficient contains three parameters. Two are used in the SNR calculation as described below. The third parameter is a flag to mark bad samples. A value of 0 marks good data, and 1 marks bad data. The first version of sample data does not have this implemented correctly.

Calculating noise equivalent radiance

The noise values are not stored directly in the file, but they can be calculated using a few fields in the file and the following formula for the noise equivalent radiance:

$$NEN = \frac{MaxMS}{100} \cdot \sqrt{\left|\frac{100 \cdot N}{MaxMS}\right| \cdot C_{photon}^2 + C_{background}^2}$$

where N is the radiance value (found in SoundingMeasurements), MaxMS is the maximum measurable signal per band (see Table 7-1), C_{photon} is the first coefficient of InstrumentHeader/snr_coef (zero-based indices [0,*,*,*]), and $C_{background}$ is the second coefficient of InstrumentHeader/snr_coef ([1,*,*,*]).

Calculating SNR

The signal to noise ratio can be calculated using the following formula:

$$SNR = \sqrt{\frac{100\, N^2}{MaxMS * \left(C_{background}^2 \frac{MaxMS}{100} + C_{photon}^2 N\right)}}$$

where N, MaxMS, C_{photon}, and $C_{background}$ are as defined above. The third entry of InstrumentHeader/snr_coef (zero-based indices [2,*,*,*]) is used to identify bad samples that should be excluded by retrieval algorithms. The third entry will be 1 if the sample is bad and 0 if it is good.

Table 7-1. *Maximum measurable signal per band.*

Band	MaxMS value (photons/m²/sr/µm)
O_2 A-band	$7.00*10^{20}$
Weak CO_2	$2.45*10^{20}$
Strong CO_2	$1.25*10^{20}$

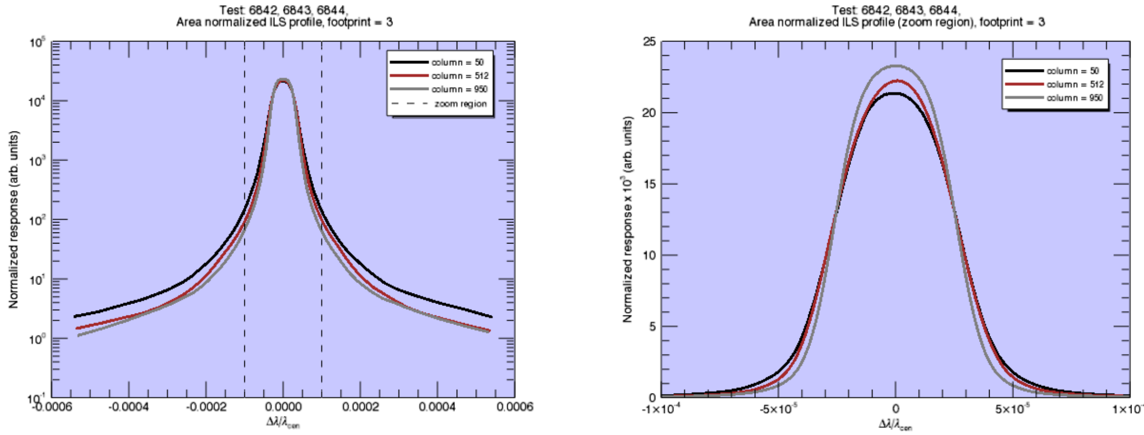

Figure 7-2. *An example of the instrument line shapes.*

7.1.3.3 ils_relative_response and ils_delta_lambda

For each band, footprint, and wavelength (3 x 8 x 1016), there are two 200-element lookup tables: ils_delta_lambda and ils_relative_response. The details of the ILS pre-launch determination are reported in Maxwell et al. (2014). These curves describe the response of each sample of the instrument, and can be used to convolve high spectral resolution spectra to spectra on the instrument resolution. Figure 7-2 illustrates one of the instrument line shapes (ILS).

8 Full Data Tables

The tables below give a full listing of the variables in each folder/group and a short description including the data type. We suggest that users familiarize themselves with the data dimensions by browsing with hdfview or a similar tool. The most useful global attributes present in all files are presented in Section 8.1. Tables for L2 data products are in Section 8.2. The groups common to L1bSc and IMAP_DOAS data products are described in Section 8.3. Finally, Section 8.4 provides a list of the data fields in each L1bSc group in the OCO-2 HDF data files.

8.1 Metadata in all Product Files

Metadata (see Table 8-1 below) contains information about the orbit, including type, start and stop times, number of frames acquired, color slice locations that are common to all of the data in the file. Note that the metadata are slightly different for each product file, as noted in the description field.

Table 8-1. *Orbit metadata common to all of the data in the file.*

Name	Type	Description
ARPAncillaryDatasetDescriptor	String	The name of the ARP file used to calibrate the data in the file.
EquatorCrossingLongitude	Float32	The interpolated longitude of the equator crossing of the OCO-2 spacecraft nadir track in the descending direction.
EquatorCrossingTime	String	The interpolated time of the equator crossing of the OCO-2 spacecraft nadir track in the descending direction.
EquatorCrossingDate	String	The date of the equator crossing of the OCO-2 spacecraft nadir track in the descending direction.
AscendingEquator CrossingLongitude	Float32	The interpolated longitude of the equator crossing of the OCO-2 spacecraft nadir track in the ascending direction.
AscendingEquator CrossingTime	String	The interpolated time of the equator crossing of the OCO-2 spacecraft nadir track in the ascending direction.
AscendingEquator CrossingDate	String	The date of the equator crossing of the OCO-2 spacecraft nadir track in the ascending direction.
OrbitStartLongitude	Float32	The interpolated longitude of the equator crossing of the OCO spacecraft nadir track in the descending direction.
OrbitStartTime	String	The interpolated time of the equator crossing of the OCO spacecraft nadir track in the descending direction.
OrbitStartDate	String	The date of the equator crossing of the OCO spacecraft nadir track in the descending direction.
OrbitEccentricity	Float32	The eccentricity of the spacecraft orbital path.
OrbitInclination	Float32	The angle between the plane of the spacecraft's orbital path and the earth's equatorial plane.
OrbitSemiMajorAxis	Float32	The length of the semimajor axis of the spacecraft orbit.
OrbitPeriod	Float32	The time span between two consecutive descending node crossings in the spacecraft orbital path.
EphemerisType	String	A character string that identifies the source of the spacecraft ephemeris data that were utilized to generate this data file.
OrbitParametersPointer	String	A pointer to one data granule or a set of data granules that provide the orbit parameters that are used to generate the data in this product.
BadPixelMapVersionNum	UInt32	Version number of corresponding bad pixel map.

Name	Type	Description
ExpectedFrames	Int32	Nominal number of frames in the product.
ActualFrames	Int32	Actual number of frames reported in the product.
FirstSoundingId	Int64	The id of the first sounding in the file.
LastSoundingId	Int64	The id of the last sounding in the file.
ReportedSoundings	Int8	Indicates the inclusion of each footprint in the data: • 1=included • 0=not included
InitialUnusedSpatialPixels	Int16	Distance in spatial pixels of the start of first footprint from edge of FPA.
SciToFPAColorOffset	Int16	Specifies the index of the first spectral pixel of arrays with FPAColor Shape that appears in the first spectral element of arrays with SciColor Shape.
ColorSlicePositionO2	Int16	Absolute spectral position of each color slice (ABO2).
ColorSlicePositionWeakCO2	Int16	Absolute spectral position of each color slice (WCO2).
ColorSlicePositionStrongCO2	Int16	Absolute spectral position of each color slice (SCO2).
L1BAlgorithmDescriptor	String	A set of character strings which list important details about the implementation of the CalApp PGE algorithms that were used to generate this product.
SpectralChannel	String	An array of character strings that identify each of the spectral channels in a sounding.
AcquisitionMode	String	The instrument mode in which the data in the product were collected. Valid values are: 'Glint', 'Nadir', 'Target', 'Sample Dark Calibration', 'Sample Lamp Calibration', 'Sample Solar/limb Calibration', 'Single-Pixel Dark Calibration', 'Single-Pixel Lamp Calibration', 'Single-Pixel Solar/limb Calibration.'
OperationMode	String	The two-letter abbreviation of the AcquisitionMode: GL, ND, TG, DS, LS, SS, BS, NP, GP, TP, DP, LP, SP, BP, XS, XP, MS, MP, SB.
ModeCounter	String	nth occurrence of this particular mode for this orbit, indicated by letter ('a', 'b', 'c', 'd', etc.).
GapStartFrame	Int64	The frame_id of the frame immediately before the gap.
GapStopFrame	Int64	The frame_id of the frame immediately after the gap.
DiffuserPosition	Float32	The position of the diffuser at the beginning of the mode.
L2IDPAlgorithmDescriptor	String	A set of character strings which list important details about the implementation of the L2 IMAP-DOAS Preprocessing PGE algorithms that were used to generate this product. L2IDP product only
OrbitParametersPointer	String	A pointer to one data granule or a set of data granules that provide the orbit parameters that are used to generate the data in this product. L2Std and L2Dia only.
ActualRetrievals	Int32	Actual number of retrievals reported in the product. L2Std and L2Dia only.
ActualGoodRetrievals	Int32	Actual number of reported retrievals with a "Good" quality flag. L2Std and L2Dia only.
RetrievalIterationLimit	Int32	Maximum number of iterations allowed in the L2 retrieval algorithm. L2Std and L2Dia only.
RadianceConversionFactor	Float32	Multiplicative factor used to convert W m^{-2} sr^{-1} um^{-1} to Ph sec^{-1} m^{-2} sr^{-1} um^{-1} at 760 nm (38.228×10^{17}). L2Std and L2Dia only.

Name	Type	Description
AbscoCO2Scale	Float32	Scale factor for the line intensities in the CO2 ABSCO tables. One factor for each band. L2Dia only.
AbscoH2OScale	Float32	Scale factor for the line intensities in the H2O ABSCO tables. One factor for each band. L2Dia only.
AbscoO2Scale	Float32	Scale factor for the line intensities in the O2 ABSCO tables. One factor for each band. L2Dia only.
L2FullPhysicsAlgorithmDescriptor	String	A set of character strings which list important details about the implementation of the Full-physics algorithm that was used to generate this product. L2Std and L2Dia only.
L2FullPhysicsInputPointer	String	List of L2 Full-physics algorithm input files. L2Std and L2Dia only.
L2FullPhysicsDataVersion	String	Mnemonic indicating the version/reprocessing status of the data used by the Full-physics algorithm. "r01" indicates original processing. "r02", "r03", etc., indicates reprocessed data. L2Std and L2Dia only.
L2FullPhysicsExeVersion	String	Indicates the build version number of the Full-physics algorithm used. Applicable to production processing only. L2Std and L2Dia only.

8.2 L2Std and L2Dia Data Tables

The L2 data products are described in Table 8-2 through Table 8-10. Recall that the L2Std variables are generally a subset of data in L2Dia. Shapes are not noted on these tables but can be easily found inside the hdf files using the dimension and shape information.

Table 8-2. *L1bSc sounding reference.*

Name	Type	Description
sounding_id_l1b	Int64	Unique identifier for each complete sounding.
sounding_qual_flag	UInt64	Quality bits specific to each sounding.
packaging_qual_flag	UInt8	Flags recording errors during packaging of L2 Full-physics and preprocessing output into retrieval arrays.
retrieval_index	Int32	The index into the Retrieval dimension of arrays in the RetrievalResults, RetrievedStateVector, and RetrievedSpectra groups for soundings associated with retrievals.

Table 8-3. *RetrievalHeader.*

Name	Type	Description
sounding_id	Int64	The sounding_id of the sounding containing the spectra used to perform the retrieval.
frame_index	Int32	Index of the frame dimension of the corresponding sounding in SoundingHeader data elements.
sounding_index	Int32	Index of the sounding dimension of the corresponding sounding in the SoundingHeader data elements.
warn_level	Int8	Provides an indication of the data quality, on scale of 0 to 19. The smaller value warn levels are expected to be of higher quality.
retrieval_time_string	String	Representative measurement time of the data used in the retrieval.
retrieval_time_tai93	Float64	Representative measurement time of the data used in the retrieval, in seconds since Jan. 1, 1993.

Table 8-4. *RetrievalGeometry*.

Name	Type	Description
retrieval_latitude_geoid	Float32	Geodetic latitude of the sounding based on standard geoid.
retrieval_longitude_geoid	Float32	Longitude of the IFOV based on standard geoid.
retrieval_latitude	Float32	Geodetic latitude of the IFOV based on SRTM Earth topography.
retrieval_longitude	Float32	Longitude of the sounding based on SRTM Earth topography.
retrieval_altitude	Float32	Altitude of the IFOV based on SRTM Earth topography.
retrieval_altitude_uncert	Float32	Standard deviation of the measure of altitude for the IFOV.
retrieval_slope	Float32	Representative slope of surface at the location of the IFOV.
retrieval_plane_fit_quality	Float32	Returns a goodness of fit. Currently this is implemented as the standard deviation of the points, to which the plane is fitted.
retrieval_aspect	Float32	Orientation of the surface slope relative to the ground track.
retrieval_surface_roughness	Float32	Standard deviation of the surface slope within the region of the IFOV.
retrieval_solar_distance	Float64	Distance between observed surface and the Sun.
retrieval_solar_azimuth	Float32	Angle between due North and the projection of the solar angle onto the Earth at the IFOV location based on topography.
retrieval_solar_zenith	Float32	Angle between the normal to the Earth geoid and the solar angle at the IFOV location based on topography.
retrieval_land_water_indicator	Int8	Flag values indicate the type of surface at the location of the sounding, such as land versus water. 0 -> Land 1 -> Ocean 2 -> Inland water 3 -> Mixed land water

Table 8-5. *PreprocessingResults data.*

Element	Type	Description
albedo_o2_abp	Float32	O_2 albedo at 785 nm and 755 nm.
dispersion_multiplier_abp	Float32	A scaling of the instrument wavelength scale to best match the modeled line positions to the measured line postitions.
reduced_chi_squared_o2_abp	Float32	O_2 reduced χ^2 retrieved by ABO2 preprocessing.
reduced_chi_squared_o2_threshold_abp	Float32	Threshold of O_2 reduced χ^2 used to set cloud_flag.
noise_o2_abp	Float32	O_2 measurement noise retrieved by ABO2 preprocessing.
signal_o2_abp	Float32	O_2 measurement signal level retrieved by ABO2 preprocessing.
snr_o2_abp	Float32	O_2 measurement SNR retrieved by ABO2 preprocessing.
surface_pressure_apriori_abp	Float32	A priori surface pressure used by ABO2 preprocessing.
surface_pressure_offset_abp	Float32	An offset term to account for imperfect spectroscopy in the ABO2 algorithm.
surface_pressure_abp	Float32	Surface pressure retrieved by ABO2 preprocessing.
surface_pressure_delta_abp	Float32	The value of surface_pressure_abp, minus the values of surface_pressure_apriori_abp and surface_pressure_offset_abp.
temperature_offset_abp	Float32	Retrieved offset for temperature profile from ABO2 algorithm.
dry_air_column_apriori_idp	Float32	Integrated vertical column of dry airmass derived from meteorological data.

Element	Type	Description
co2_column_weak_band_idp	Float32	CO_2 vertical column density (from WCO2 band).
co2_column_weak_band_apriori_idp	Float32	A priori CO_2 vertical column density from ECMWF forecast.
co2_column_weak_band_uncert_idp	Float32	1-sigma error in the CO_2 vertical column density (from WCO2 band).
co2_column_strong_band_idp	Float32	CO_2 vertical column density (from SCO2 band).
co2_column_strong_band_apriori_idp	Float32	A priori CO_2 vertical column density from ECMWF forecast.
co2_column_strong_band_uncert_idp	Float32	1-sigma error in the CO_2 vertical column density (from SCO2 band).
co2_weak_band_processing_flag_idp	Int8	Flag indicating whether the WCO2 analysis succeeded: • 0 = "Processing succeeded" • 1 = "Processing failed" • 2 = "Processing skipped" All other values undefined.
co2_strong_band_processing_flag_idp	Int8	Flag indicating whether the SCO2 analysis succeeded: • 0 = "Processing succeeded" • 1 = "Processing failed" • 2 = "Processing skipped" All other values undefined.
cloud_flag_idp	Int8	Cloud flag derived from IMAP-DOAS algorithm.
co2_ratio_idp	Float32	Ratio of retrieved CO_2 column (no scattering code) in WCO2 and SCO2 bands.
h2o_ratio_idp	Float32	Ratio of retrieved H_2O column (no scattering code) in WCO2 and SCO2 bands.
h2o_ratio_uncert_idp	Float32	1-sigma error in the ratio of retrieved H_2O column (no scattering code) in WCO2 and SCO2 bands.
o2_ratio_idp	Float32	Ratio of retrieved and ECMWF O_2 column
fluorescence_qual_flag_idp	UInt8	Quality flag on the output of the IMAP-DOAS fluorescence retrieval
fluorescence_offset_relative_757nm_idp	Float32	Fraction of continuum level radiance explained by an additive offset term in the 757 nm spectral window (unitless).
fluorescence_offset_relative_771nm_idp	Float32	As above for 711nm window.
fluorescence_radiance_757nm_idp	Float32	Radiance generated by fluorescence at 755nm.
fluorescence_radiance_757nm_uncert_idp	Float32	Standard deviation of the radiance generated by fluorescence at 755nm.
fluorescence_radiance_771nm_idp	Float32	Radiance generated by fluorescence at 771 nm.
fluorescence_radiance_771nm_uncert_idp	Float32	Standard deviation of the radiance generated by fluorescence at 771 nm.
selection_priority	Int8	Indicator of the likelihood of generating a good retrieval from the sounding. 0 = most likely, 20 = least likely.
cloud_flag_abp	Int8	Cloud flag derived from O_2 A-band preprocessing algorithm. • -2 = "Measurement unusable" • -1 = "Did not converge" • 0 = "Definitely cloudy" • 1 = "Probably cloudy" • 2 = "Probably clear" • 3 = "Very clear" All other values undefined.

Table 8-6. *RetrievalResults data.*

Element	Type	Units	Description
num_active_levels	Int16		Number of levels in atmospheric model.
surface_type	String		Type of model used for the Earth's surface. Valid values: "Lambertian" or "Coxmunk,Lambertian."
outcome_flag	Int8		Flag providing details of processing results of L2 Full-physics retrieval: • 1 = Passed internal quality check • 2 = Failed internal quality check • 3 = Reached maximum allowed iterations • 4 = Reached maximum allowed divergences
vector_pressure_levels_ecmwf	Float32	Pa	Pressure altitude corresponding to each ECMWF atmospheric level.
temperature_profile_ecmwf	Float32	K	ECMWF temperature profile interpolated to observation location, time.
specific_humidity_profile_ecmwf	Float32	kg/kg	ECMWF specific humidity profile interpolated to observation location, time.
surface_pressure_fph	Float32	Pa	Surface pressure retrieved by Full-physics algorithm.
surface_pressure_apriori_fph	Float32	Pa	A priori surface pressure retrieved by Full-physics algorithm.
surface_pressure_uncert_fph	Float32	Pa	Uncertainty in the surface pressure retrieved by the Full-physics algorithm.
vector_pressure_levels	Float32	Pa	Pressure altitude corresponding to each atmospheric level.
vector_pressure_levels_apriori	Float32	Pa	A priori pressure altitude corresponding to each atmospheric level.
diverging_steps	Int16		Number of iterations in which solution diverged.
iterations	Int16		Number of iterations.
dof_co2_profile	Float32		Degrees of freedom (X_{CO2} only).
dof_full_vector	Float32		Degrees of freedom (Full state vector).
xco2	Float32	mole/mole	Column-averaged CO_2 dry air mole fraction.
xco2_apriori	Float32	mole/mole	A priori of column-averaged CO_2 dry air mole fraction.
xco2_uncert	Float32	mole/mole	Error in column averaged CO_2 dry air mole fraction.
xco2_uncert_noise	Float32	mole/mole	Variance of CO_2 due to noise.
xco2_uncert_smooth	Float32	mole/mole	Variance of CO_2 due to smoothing.
xco2_uncert_interf	Float32	mole/mole	Variance of CO_2 due to interference.
co2_profile	Float32	mole/mole	Vertical profile of CO_2.
co2_profile_apriori	Float32	mole/mole	Vertical a priori profile of CO_2.
co2_profile_uncert	Float32	mole/mole	Vertical profile of CO_2 uncertainty.
xco2_pressure_weighting_function	Float32		Pressure weighting function used to form X_{CO2}.
xco2_avg_kernel	Float32		Column averaging kernel.
xco2_avg_kernel_norm	Float32		Normalized column averaging kernel.
co2_profile_averaging_kernel_matrix	Float32		Averaging kernel for CO_2 profile.
co2_profile_covariance_matrix	Float32	$mole^2/mole^2$	Covariance matrix for CO_2 profile.
wind_speed	Float32	m/s	Retrieved Cox-Munk wind speed.

Element	Type	Units	Description
wind_speed_apriori	Float32	m/s	A priori of retrieved Cox-Munk wind speed.
wind_speed_uncert	Float32	m/s	Uncertainty of retrieved Cox-Munk wind speed.
h2o_scale_factor	Float32		Retrieved scale factor for H_2O profile.
h2o_scale_factor_apriori	Float32		A priori of retrieved scale factor for H_2O profile.
h2o_scale_factor_uncert	Float32		Uncertainty of retrieved scale factor for H_2O profile.
temperature_offset_fph	Float32	K	Retrieved offset of temperature profile.
temperature_offset_apriori_fph	Float32	K	A priori of retrieved offset of temperature profile.
temperature_offset_uncert_fph	Float32	K	Uncertainty of retrieved offset of temperature profile.
fluorescence_at_reference	Float32	Ph/s/m^2/sr/µm	Retrieved fluorescence at 0.757 microns.
fluorescence_at_reference_apriori	Float32	Ph/s/m^2/sr/µm	A priori of retrieved fluorescence at 0.757 microns.
fluorescence_at_reference_uncert	Float32	Ph/s/m^2/sr/µm	Uncertainty of retrieved fluorescence at 0.757 microns.
fluorescence_slope	Float32	Ph/s/m^2/sr/µm^2	Retrieved fluorescence slope at 0.757 microns.
fluorescence_slope_apriori	Float32	Ph/s/m^2/sr/µm^2	A priori of retrieved fluorescence slope at 0.757 microns.
fluorescence_slope_uncert	Float32	Ph/s/m^2/sr/µm^2	Uncertainty of retrieved fluorescence slope at 0.757 microns.
eof_1_scale_o2	Float32		Retrieved scale factor of first empirical orthogonal residual function in O2 channel.
eof_1_scale_apriori_o2	Float32		A priori of retrieved scale factor of first empirical orthogonal residual function in O2 channel.
eof_1_scale_uncert_o2	Float32		Uncertainty of retrieved scale factor of first empirical orthogonal residual function in O2 channel.
eof_1_scale_weak_co2	Float32		Retrieved scale factor of first empirical orthogonal residual function in WCO2 channel.
eof_1_scale_apriori_weak_co2	Float32		A priori of retrieved scale factor of first empirical orthogonal residual function in WCO2 channel.
eof_1_scale_uncert_weak_co2	Float32		Uncertainty of retrieved scale factor of first empirical orthogonal residual function in WCO2 channel.
eof_1_scale_strong_co2	Float32		Retrieved scale factor of first empirical orthogonal residual function in SCO2 channel.
eof_1_scale_apriori_strong_co2	Float32		A priori of retrieved scale factor of first empirical orthogonal residual function in SCO2 channel.
eof_1_scale_uncert_strong_co2	Float32		Uncertainty of retrieved scale factor of first empirical orthogonal residual function in SCO2 channel.
retrieved_dry_air _column_layer_thickness	Float32	molecules/m^2	Retrieved vertical column of dry air per atmospheric layer.
retrieved_wet_air_column _layer_thickness	Float32	molecules/m^2	Retrieved vertical column of wet air per atmospheric layer.
retrieved_h2o _column_layer_thickness	Float32	molecules/m^2	Retrieved vertical column of H_2O per atmospheric layer.
apriori_o2_column	Float32	molecules/m^2	A priori vertical column of O_2.

Element	Type	Units	Description
retrieved_co2_column	Float32	molecules/m^2	Retrieved vertical column of CO_2.
retrieved_h2o_column	Float32	molecules/m^2	Retrieved vertical column of H_2O.
retrieved_o2_column	Float32	molecules/m^2	Retrieved vertical column of O_2.
last_step_levenberg_marquardt _parameter	Float32		Levenberg Marquardt parameter corresponding to last iteration.
	Int16		Number of levels in atmospheric model.

Table 8-7. *AlbedoResults data.*

Element	Type	Units	Description
albedo_o2_fph	Float32		Retrieved Lambertian component of albedo at 0.77 microns.
albedo_weak_co2_fph	Float32		Retrieved Lambertian component of albedo at 1.615 microns.
albedo_strong_co2_fph	Float32		Retrieved Lambertian component of albedo at 2.06 microns.
albedo_apriori_o2_fph	Float32		A priori of retrieved Lambertian component of albedo at 0.77 microns.
albedo_apriori_weak_co2_fph	Float32		A priori of retrieved Lambertian component of albedo at 1.615 microns.
albedo_aprioiri_strong_co2_fph	Float32		A priori of retrieved Lambertian component of albedo at 2.06 microns.
albedo_uncert_o2_fph	Float32		Uncertainty of retrieved Lambertian component of albedo 0.77 microns.
albedo_uncert_weak_co2_fph	Float32		Uncertainty of retrieved Lambertian component of albedo at 1.615 microns.
albedo_uncert_strong_co2_fph	Float32		Uncertainty of retrieved Lambertian component of albedo at 2.06 microns.
albedo_slope_o2	Float32	cm	Retrieved spectral dependence of Lambertian component of albedo within O2 channel.
albedo_slope_weak_co2	Float32	cm	Retrieved spectral dependence of Lambertian component of albedo within WCO2 channel.
albedo_slope_strong_co2	Float32	cm	Retrieved spectral dependence of Lambertian component of albedo within SCO2 channel.
albedo_slope_apriori_o2	Float32	cm	A priori of retrieved spectral dependence of Lambertian component of albedo within O2 channel.
albedo_slope_apriori_weak_co2	Float32	cm	A priori of retrieved spectral dependence of Lambertian component of albedo within WCO2 channel.
albedo_slope_apriori_strong_co2	Float32	cm	A priori of spectral dependence of Lambertian component of albedo within SCO2 channel.
albedo_slope_uncert_o2	Float32	cm	Uncertainty of retrieved spectral dependence of Lambertian component of albedo within O2 channel.
albedo_slope_uncert_weak_co2	Float32	cm	Uncertainty of retrieved spectral dependence of Lambertian component of albedo within WCO2 channel.
albedo_slope_uncert_strong_co2	Float32	cm	Uncertainty of spectral dependence of Lambertian component of albedo within SCO2 channel.

Table 8-8. *DispersionResults data.*

Element	Type	Description
dispersion_offset_o2	Float64	Retrieved dispersion offset term in O2 channel.
dispersion_offset_weak_co2	Float64	Retrieved dispersion offset term in WCO2 channel.
dispersion_offset_strong_co2	Float64	Retrieved dispersion offset term in SCO2 channel.
dispersion_offset_apriori_o2	Float64	A priori of retrieved dispersion offset term in O2 channel.
dispersion_offset_apriori_weak_co2	Float64	A priori of retrieved dispersion offset term in WCO2 channel.
dispersion_offset_apriori_strong_co2	Float64	A priori of retrieved dispersion offset term in SCO2 channel.
dispersion_offset_uncert_o2	Float32	Uncertainty of retrieved dispersion offset term in O2 channel.
dispersion_offset_uncert_weak_co2	Float32	Uncertainty of retrieved dispersion offset term in WCO2 channel.
dispersion_offset_uncert_strong_co2	Float32	Uncertainty of retrieved dispersion offset term in SCO2 channel.

Table 8-9. *AerosolResults data.*

Element	Type	Description
aerosol_1_gaussian_log_param	Float32	Retrieved gaussian log parameters for first aerosol type [total log aod, center pressure/surf-pressure, pressure sigma/surf-pressure]
aerosol_1_gaussian_log_param_apriori	Float32	Apriori of retrieved gaussian log parameters for first aerosol type
aerosol_1_gaussian_log_param_uncert	Float32	Uncertainty of retrieved gaussian log parameters for first aerosol type
aerosol_2_gaussian_log_param	Float32	Retrieved gaussian log parameters for second aerosol type [total log aod, center pressure/surf-pressure, pressure sigma/surf-pressure]
aerosol_2_gaussian_log_param_apriori	Float32	Apriori of retrieved gaussian log parameters for second aerosol type
aerosol_2_gaussian_log_param_uncert	Float32	Uncertainty of retrieved gaussian log parameters for second aerosol type
aerosol_3_gaussian_log_param	Float32	Retrieved gaussian log parameters for water aerosol type [total log aod, center pressure/surf-pressure, pressure sigma/surf-pressure]
aerosol_3_gaussian_log_param_apriori	Float32	Apriori of retrieved gaussian log parameters for water aerosol type
aerosol_3_gaussian_log_param_uncert	Float32	Uncertainty of retrieved gaussian log parameters for water aerosol type
aerosol_4_gaussian_log_param	Float32	Retrieved gaussian log parameters for ice aerosol type [total log aod, center pressure/surf-pressure, pressure sigma/surf-pressure]
aerosol_4_gaussian_log_param_apriori	Float32	Apriori of retrieved gaussian log parameters for ice aerosol type
aerosol_1_aod	Float32	Retrieved total column-integrated aerosol optical depth for first aerosol type
aerosol_1_aod_low	Float32	Retrieved column-integrated aerosol optical depth for first aerosol type for pressure levels greater than 80,000 Pa
aerosol_1_aod_mid	Float32	Retrieved column-integrated aerosol optical depth for first aerosol type for pressure levels between 50,000 and 80,000 Pa
aerosol_1_aod_high	Float32	Retrieved column-integrated aerosol optical depth for first aerosol type for pressure levels less than 50,000 Pa

Element	Type	Description
aerosol_2_aod	Float32	Retrieved total column-integrated aerosol optical depth for second aerosol type
aerosol_2_aod_low	Float32	Retrieved column-integrated aerosol optical depth for second aerosol type for pressure levels greater than 80,000 Pa
aerosol_2_aod_mid	Float32	Retrieved column-integrated aerosol optical depth for second aerosol type for pressure levels between 50,000 and 80,000 Pa
aerosol_2_aod_high	Float32	Retrieved column-integrated aerosol optical depth for second aerosol type for pressure levels less than 50,000 Pa
aerosol_3_aod	Float32	Retrieved total column-integrated aerosol optical depth for water aerosol type
aerosol_3_aod_low	Float32	Retrieved column-integrated aerosol optical depth for water aerosol type for pressure levels greater than 80000 Pa
aerosol_3_aod_mid	Float32	Retrieved column-integrated aerosol optical depth for water aerosol type for pressure levels between 50,000 and 80,000 Pa
aerosol_3_aod_high	Float32	Retrieved column-integrated aerosol optical depth for water aerosol type for pressure levels less than 50,000 Pa
aerosol_4_aod	Float32	Retrieved total column-integrated aerosol optical depth for ice aerosol type
aerosol_4_aod_low	Float32	Retrieved column-integrated aerosol optical depth for ice aerosol type for pressure levels greater than 80,000 Pa
aerosol_4_aod_mid	Float32	Retrieved column-integrated aerosol optical depth for ice aerosol type for pressure levels between 50,000 and 80,000 Pa
aerosol_4_aod_high	Float32	Retrieved column-integrated aerosol optical depth for ice aerosol type for pressure levels less than 50,000 Pa
aerosol_types	String	Retrieved aerosol type
aerosol_total_aod	Float32	Retrieved total column-integrated aerosol optical depth for all aerosol types.
aerosol_total_aod_low	Float32	Retrieved column-integrated aerosol optical depth for all aerosol types for pressure levels greater than 80,000 Pa.
aerosol_total_aod_mid	Float32	Retrieved column-integrated aerosol optical depth for all aerosol types for pressure levels between 50,000 and 80,000 Pa.
aerosol_total_aod_high	Float32	Retrieved column-integrated aerosol optical depth for all aerosol types for pressure levels less than 50,000 Pa.

Table 8-10. *SpectralParameters data.*

Element	Type	Description
residual_mean_square_o2	Float32	Mean of the squares of the residuals for the O2 band.
residual_mean_square_weak_co2	Float32	Mean of the squares of the residuals for the WCO2 band.
residual_mean_square_strong_co2	Float32	Mean of the squares of the residuals for the SCO2 band.
signal_o2_fph	Float32	Aggregate signal in the O2 band.
signal_weak_co2_fph	Float32	Aggregate signal in the WCO2 band.
signal_strong_co2_fph	Float32	Aggregate signal in the SCO2 band.
noise_o2_fph	Float32	Aggregate noise in the O2 band.
noise_weak_co2_fph	Float32	Aggregate noise in the WCO2 band.
noise_strong_co2_fph	Float32	Aggregate noise in the SO2 band.

Element	Type	Description
relative_residual_mean_square_o2	Float32	Mean square of the residuals divided by the signal for the ABO2 band.
relative_residual_mean_square_weak_co2	Float32	Mean square of the residuals divided by the signal for the WCO2 band.
relative_residual_mean_square_strong_co2	Float32	Mean square of the residuals divided by the signal for the SCO2 band.
reduced_chi_squared_o2_fph	Float32	Reduced χ^2 of spectral fit of the O2 band.
reduced_chi_squared_weak_co2_fph	Float32	Reduced χ^2 of spectral fit of the WCO2 band.
reduced_chi_squared_strong_co2_fph	Float32	Reduced χ^2 of spectral fit of the SCO2 band.
measured_radiance		
measured_radiance_uncert		
modeled_radiance		
wavelength		

8.3 L1b and IMAP-DOAS Data Tables

The primary dimension of the data is the sounding. For each sounding, there are eight spatial footprints. For each footprint, there are three spectral bands: O_2 A-band, Weak CO_2, and Strong CO_2.

8.3.1 FrameGeometry

The FrameGeometry group (see Table 8-11 below) contains detailed information about the spacecraft position and orientation during the observations.

Table 8-11. *Spacecraft position and orientation during observations.*

Name	Type	Description
spacecraft_position	Float32	Interpolated spacecraft position for cross track row.
spacecraft_velocity	Float32	Interpolated spacecraft velocity for cross track row.
roll	Float32	Representative spacecraft attitude for cross track row.
pitch	Float32	Representative spacecraft attitude for cross track row.
yaw	Float32	Representative spacecraft attitude for cross track row.
spacecraft_lat	Float32	Geodetic latitude of nadir track position of spacecraft when the IFOV was acquired.
spacecraft_lon	Float32	Longitude of position of the spacecraft when the sounding was acquired.
spacecraft_alt	Float32	Altitude of the spacecraft above the reference ellipsoid when the IFOV was acquired.
relative_velocity	Float32	Relative SC/Target motion speed in the inertial reference projected aligned with LOS.
ground_track	Float32	Subsatellite ground track orientation relative to the local north.

8.3.2 SoundingGeometry

The SoundingGeometry group (see Table 8-12 below) contains detailed information about the geometric location, atmospheric geometry, and surface conditions for the sounding which combines the three spectrometers.

Table 8-12. *Geometric location, atmospheric geometry, and surface conditions.*

Name	Type	Description
sounding_id	Int64	Unique identifier for each complete sounding.
sounding_time_string	String	Representative measurement time of the sounding.

Name	Type	Description
sounding_time_tai93	Float64	Representative measurement time of the sounding in seconds since Jan. 1, 1993.
sounding_overlap	Float32	Union in area of all three spectrum footprints relative to average area of all three spectrum footprints.
sounding_overlap_o2_weak_co2	Float32	This is the union in area of the footprints of spectrum one and two relative to the average area of the two footprints.
sounding_overlap_weak_co2_strong_co2	Float32	This is the union in area of the footprints of spectrum two and three relative to the average area of the two footprints.
sounding_overlap_strong_co2_o2	Float32	This is the union in area of the footprints of spectrum one and three relative to the average area of the two footprints.
sounding_slant_path_diff_o2_weak_co2	Float32	The difference in slant path difference between ABO2 and WCO2 footprints.
sounding_slant_path_diff_weak_co2_strong_co2	Float32	The difference in slant path difference between WCO2 and SCO2 footprints.
sounding_slant_path_diff_strong_co2_o2	Float32	The difference in slant path difference between SCO2 and ABO2 footprints.
sounding_center_offset_o2_weak_co2	Float32	Distance between the ABO2 band footprint center and the WCO2 band footprint center.
sounding_center_offset_weak_co2_strong_co2	Float32	Distance between the WCO2 band footprint center and the SCO2 band footprint center.
sounding_center_offset_strong_co2_o2	Float32	Distance between the SCO2 band footprint center and the ABO2 band footprint center.
sounding_qual_flag	UInt64	Quality bits specific to each pixel.
sounding_latitude_geoid	Float32	Geodetic latitude of the sounding based on standard geoid.
sounding_longitude_geoid	Float32	Longitude of the IFOV based on standard geoid.
sounding_latitude	Float32	Geodetic latitude of the IFOV based on SRTM Earth topography.
sounding_longitude	Float32	Longitude of the sounding based on SRTM Earth topography.
sounding_altitude	Float32	Altitude of the IFOV based on SRTM Earth topography.
sounding_altitude_uncert	Float32	Standard deviation of the measure of altitude for the IFOV.
sounding_slope	Float32	Representative slope of surface at the location of the IFOV.
sounding_plane_fit_quality	Float32	Returns a goodness of fit. Currently the goodness-of-fit is implemented as the standard deviation of the points, to which the plane is fitted, with the expected values taken as the orthogonal projection of the points onto the plane.
sounding_aspect	Float32	Orientation of the surface slope relative to the ground track.
sounding_surface_roughness	Float32	Standard deviation of the surface slope within the region of the IFOV.
sounding_solar_distance	Float64	Distance between observed surface and the Sun.

Name	Type	Description
sounding_solar_azimuth	Float32	Angle between due North and the projection of the solar angle onto the Earth at the IFOV location based on topography.
sounding_solar_zenith	Float32	Angle between the normal to the Earth geoid and the solar angle at the IFOV location based on topography.
sounding_azimuth	Float32	Angle between due North and the projection of the view vector toward the instantaneous position of the spacecraft from the IFOV location based on topography.
sounding_zenith	Float32	The angle between the normal to the Earth geoid and the view vector that extends from the center of the sounding toward the instantaneous position of the spacecraft. The location of the sounding center is based upon a Digital Elevation Model (DEM) of the Earth's surface.
sounding_land_water_indicator	Int8	Flag values indicate the type of surface at the location of the sounding, such as land versus water: • 0 = Land • 1 = Ocean • 2 = Inland water • 3 = Mixed land water

8.4 L1b Data Tables

8.4.1 SoundingMeasurements

The SoundingMeasurements group (see Table 8-13. *Calibrated radiance spectra.*Table 8-13 below) contains the calibrated radiance spectra. The spectral data are stored as 32 bit floats whose units are photons/m^2/sr/ µm.

Table 8-13. *Calibrated radiance spectra.*

Name	Description
radiance_o2	Spectra (ABO2).
radiance_weak_co2	Spectra (WCO2).
radiance_strong_co2	Spectra (SCO2).

8.4.2 SliceMeasurements

The SliceMeasurements group (see Table 8-14 below) contains the calibrated radiance values for the color slices. The spectral slice data are stored as 32 bit floats whose units are photons/m^2/sr/ µm.

Table 8-14. *Calibrated radiance values for the color slices.*

Name	Description
radiance_slice_o2	Radiance values for all slice pixels (ABO2).
radiance_slice_weak_co2	Radiance values for all slice pixels (WCO2).
radiance_slice_strong_co2	Radiance values for all slice pixels (SCO2).

8.4.3 FootprintGeometry

The FootprintGeometry group (see Table 8-15 below) contains detailed information about the location and observational geometry for each focal plane and spatial footprint.

Table 8-15. *Location and observational geometry for each focal plane and spatial footprint.*

Name	Type	Description
footprint_time_tai93	Float64	Seconds since Jan. 1, 1993.
footprint_time_string	String	Time stamp associated with the center of footprint.
footprint_o2_qual_flag	UInt16	
footprint_weak_co2_qual_flag	UInt16	
footprint_strong_co2_qual_flag	UInt16	
footprint_latitude_geoid	Float32	Geodetic latitude of the sounding based on standard geoid.
footprint_longitude_geoid	Float32	Longitude of the IFOV based on standard geoid.
footprint_latitude	Float32	DEM-based latitude of the footprint center.
footprint_longitude	Float32	DEM-based longitude of the footprint center.
footprint_altitude	Float32	DEM-based altitude of the footprint center.
footprint_altitude_uncert	Float32	Standard deviation of the measure of altitude for the IFOV.
footprint_slope	Float32	Representative slope of surface at the location of the IFOV.
footprint_plane_fit_quality	Float32	Returns a goodness of fit. Currently the goodness-of-fit is implemented as the standard deviation of the points, to which the plane is fitted, with the expected values taken as the orthogonal projection of the points onto the plane.
footprint_aspect	Float32	Orientation of the surface slope relative to the ground track.
footprint_surface_roughness	Float32	Standard deviation of the surface slope within the region of the IFOV.
footprint_solar_azimuth	Float32	Angle between due North and the projection of the solar angle onto the Earth at the IFOV location based on topography.
footprint_solar_zenith	Float32	Angle between the normal to the Earth geoid and the solar angle at the IFOV location based on topography.
footprint_azimuth	Float32	Angle between due North and the projection of the view vector toward the instantaneous position of the spacecraft from the IFOV location based on topography.
footprint_zenith	Float32	Angle between the normal to the Earth geoid and the view vector toward the instantaneous position of the spacecraft from the IFOV location based on topography.
footprint_vertex_longitude	Float32	DEM-based geodetic longitude of the footprint vertices.
footprint_vertex_latitude	Float32	DEM-based geodetic latitude of the footprint vertices.
footprint_vertex_altitude	Float32	DEM-based geodetic altitude of the footprint vertices.
footprint_stokes_coefficients	Float32	Undefined.

8.4.4 FrameConfiguration

The FrameConfiguration group (see Table 8-16 below) contains information about how the detectors and color slices were configured during the observations.

Table 8-16. *Configuration of detectors and color slices.*

Name	Type	Description

Name	Type	Description
color_slice_position_o2	Int16	Absolute spectral position, in pixels, of each color slice. There may be up to 20 slices.
color_slice_position_strong_co2	Int16	Absolute spectral position, in pixels, of each color slice. There may be up to 20 slices.
color_slice_position_weak_co2	Int16	Absolute spectral position, in pixels, of each color slice. There may be up to 20 slices.
footprint_spatial_end_position	Uint8	Position of end of each footprint, in pixels, relative to initial_unused_pixels.
footprint_spatial_start_position	Uint8	Position of start of each footprint, in pixels, relative to initial_unused_pixels.
initial_unused_pixels	Int16	Distance in pixels of the start of first footprint from edge of FPA.

8.4.5 FrameHeader

The FrameHeader group (see Table 8-17 below) contains the frame identification numbers, times (UTC and TAI), the L1b frame quality flag, and some details of the array clocking.

Table 8-17. *Frame identification data.*

Name	Type	Description
frame_id	Int64	The mission-unique frame identifier.
frame_time_string	String	Time of telemetry frame measure.
frame_time_tai93	Float64	Time of telemetry frame in seconds since Jan 1, 1993.
frame_qual_flag	UInt64	Frame-based quality assessment flags.
clocking_offset_start	Int32	The first pixel where the flight software applies a focal plane clocking correction. Negative value indicates the clocking shift operates downward, positive value implies clocking shift is upward.
clocking_offset_interval	Int16	The number of pixels between each successive pixel shift used to apply a clocking correction.

8.4.6 FrameTemperatures

The FrameTemperatures group (see Table 8-18 below) contains selected temperatures for each frame.

Table 8-18. *Selected temperature data.*

Name	Type	Description
temp_fpa	Float32	Ambient temperature of the three focal plane arrays.
temp_optical_bench_grating_mz	Float32	Ambient temperature of the optical bench grating.
temp_relay_sco2_mz	Float32	Ambient temperature of the relay.
temp_telescope	Float32	Ambient temperature of the telescope.
temp_shroud_py_tz1	Float32	Ambient temperature of the shroud.
temp_afe_electronics_enclosure	Float32	Ambient temperature of the AFE electronics enclosure.
temp_smooth_optical_bench_grating_mz	Float32	Optics temperature for this frame resulting from noise-reduction processing.
temp_smooth_fpa_o2	Float32	FPA temperature for this frame resulting from noise-reduction processing.
temp_smooth_fpa_weak_co2	Float32	FPA temperature for this frame resulting from noise-reduction processing.

Name	Type	Description
temp_smooth_fpa_strong_co2	Float32	FPA temperature for this frame resulting from noise-reduction processing.

8.4.7 InstrumentHeader

The InstrumentHeader group (see Table 8-19 below) contains information about the performance of the instrument.

Table 8-19. *Instrument performance data.*

Name	Type	Description
ils_delta_lambda	Float32	Wavelength offset from peak response for sampled data.
ils_relative_response	Float32	The relative response defined at ils_delta_lambda.
full_width_half_maximum	Float32	The spectral response width at full width half maximum response, per pixel.
measureable_signal_max_observed	Float32	The maximum radiance measurable by each spectrometer.
snr_coef	Float64	The SNR coefficients.
dispersion_coef_samp	Float64	Coefficients that express the relationship between the spectral element index and its associated wavelength.
residual_estimate	Float32	Empirical estimate of the systematic residuals that cannot be removed by calibration.

9 Tools and Data Services

9.1 HDFView

HDFView is a Java based graphical user interface created by the HDF Group that can be used to browse all ACOS HDF products. The utility allows users to view all objects in an HDF file hierarchy, which is represented as a tree structure. HDFView can be downloaded or support found at: http://www.hdfgroup.org/hdf-java-html/hdfview/.

9.2 Mirador

Level 2 data from OCO-2 will be available at the Goddard Earth Sciences Data and Information Services Center (GES DISC) about 7-9 months after the satellite instrument is launched. This is the pre-launch release of the Data User's Guide and is intended to provide information on data formats and volumes. This section will be updated after launch with more information about obtaining data thought the GES DISC.

The GES DISC provides basic temporal, advanced (event), and spatial searches through its search and download engine, Mirador (http://mirador.gsfc.nasa.gov). Mirador offers various download options that suit users with different preferences and different levels of technical skills. Users can start from a point where they don't know anything about these particular data, its location, size, format, etc., to quickly find what they need by just providing relevant keywords, like "OCO-2," or "CO2."

9.3 JPL CO2 Virtual Science Data Environment

The JPL CO_2 Virtual Science Data Environment (co2.jpl.nasa.gov) provides access to and information about satellite observations of carbon dioxide. The site is currently being redesigned with a new user interface and upgraded services. The expectation is that by the time of the OCO-2 launch, July 1, 2014, that site will have many improvements in place. The site is currently operational for creating gridded data product from ACOS and other satellite instruments.

9.4 Sample code for reading L2 data

9.4.1 IDL

Appendix A contains a set of code that can be used to read the hdf data products. These code examples will also be available on the JPL CO2 Virtual Science Data Environment. An example of how you would use the is presented here:

```
ReadList_L2=['RetrievalHeader/sounding_id_reference', $
        'RetrievalResults/aerosol_total_aod', $
        'RetrievalResults/albedo_o2_fph', $
        'RetrievalResults/albedo_strong_co2_fph', $
        'RetrievalResults/outcome_flag', $
        'RetrievalResults/surface_pressure_apriori_fph', $
        'RetrievalResults/surface_pressure_fph', $
        'RetrievalResults/surface_type', $
        'RetrievalResults/wind_speed', $
        'RetrievalResults/wind_speed_apriori', $
        'RetrievalResults/xco2', $
```

```
                'RetrievalResults/xco2_uncert', $
                'SpectralParameters/noise_o2_fph', $
                'SpectralParameters/noise_strong_co2_fph', $
                'SpectralParameters/noise_weak_co2_fph', $
                'SpectralParameters/signal_o2_fph', $
                'SpectralParameters/signal_strong_co2_fph', $
                'SpectralParameters/signal_weak_co2_fph', $
                'SpectralParameters/reduced_chi_squared_o2_fph', $
                'SpectralParameters/reduced_chi_squared_weak_co2_fph' $
               ]
subnames = ['retrievalheaders', 'retrievalresults', 'spectralparameters']
; the line below is used of there is only ever one file
; L2 = READ_H5_FILE (l2fullfile, READ=ReadList_L2)
; if there are multiple files, this loop handles it
NumL2Files = N_ELEMENTS (L2FullFile)
  FOR F = 0, NumL2Files - 1 DO BEGIN
 temp = READ_H5_FILE (l2FullFile[F], READ=ReadList_L2)
 IF (F eq 0) THEN BEGIN
  L2_Temp = Temp
  L2 = L2_Temp  ; in case there is only one file
 ENDIF ELSE BEGIN
  L2 = [L2_Temp, Temp]
  L2_temp = L2  ; to be ready to add to it
 ENDELSE
 ENDFOR
```

9.4.2 Python

It is also simple to use python to read and manipulate the product files. An example is provided here:

```
'''
Function read OCO-2 L2Std files in HDF 5 format.

@dependecies: h5py
@author: bwknosp

'''

import h5py

# default variables to be read out of the files
# the keys are how the variables will be named in the output returned to the user
# values are the path to the variable in the OCO-2 L2Std file
# user should edit as needed
VARIABLES = {   'co2_ratio' : '/PreprocessingResults/co2_ratio_idp',
```

```
                'h2o_ratio' : '/PreprocessingResults/h2o_ratio_idp',
            'xco2': '/RetrievalResults/xco2',
                'xco2_uncert' : '/RetrievalResults/xco2_uncert',
                'xco2_apriori' : '/RetrievalResults/xco2_apriori',
                'surface_type' : 'RetrievalResults/surface_type',
                'psurf' : '/RetrievalResults/surface_pressure_fph',
                'albedo_1' : '/AlbedoResults/albedo_o2_fph',
                'albedo_2' : '/AlbedoResults/albedo_weak_co2_fph',
                'albedo_3' : '/AlbedoResults/albedo_strong_co2_fph',
                'albedo_slope_1' : '/AlbedoResults/albedo_slope_o2',
                'albedo_slope_2' : '/AlbedoResults/albedo_slope_weak_co2',
                'albedo_slope_3' : '/AlbedoResults/albedo_slope_strong_co2',
                'aod_total' : '/AerosolResults/aerosol_total_aod',
                'deltaT' : '/RetrievalResults/temperature_offset_apriori_fph',
                'h2o_scale' : '/RetrievalResults/h2o_scale_factor',
                'psurf_apriori' : '/RetrievalResults/surface_pressure_apriori_fph',
                'chi_sq_1' : 'SpectralParameters/reduced_chi_squared_o2_fph',
                'chi_sq_2' : 'SpectralParameters/reduced_chi_squared_weak_co2_fph',
                'chi_sq_3' : 'SpectralParameters/reduced_chi_squared_strong_co2_fph',
                'pwf' : '/RetrievalResults/xco2_pressure_weighting_function',
                'co2_profile' : '/RetrievalResults/co2_profile',
                'co2_profile_apriori' : '/RetrievalResults/co2_profile_apriori',
                'xco2_ak' : '/RetrievalResults/xco2_avg_kernel',
                'latitude' : '/RetrievalGeometry/retrieval_latitude',
                'longitude' : '/RetrievalGeometry/retrieval_longitude',
                'time' : '/RetrievalHeader/retrieval_time_tai93',
                'altitude' : '/RetrievalGeometry/retrieval_altitude',
                'land_fraction' : '/RetrievalGeometry/retrieval_land_water_indicator',
                'observation_type' : '/Metadata/OperationMode',
                'solar_zenith' : '/RetrievalGeometry/retrieval_solar_zenith',
                'solar_azimuth' : '/RetrievalGeometry/retrieval_solar_azimuth',
                'satellite_zenith' : '/RetrievalGeometry/retrieval_zenith',
                'satellite_azimuth' : '/RetrievalGeometry/retrieval_azimuth',
                'sounding_id' : '/RetrievalHeader/sounding_id',
                'warn_level' : '/RetrievalHeader/warn_level' }


def readL2Std(filename):
    '''
    Function that reads the L2Std file for the variables specified in
    the VARIABLES dict and outputs a dict with the data values.

    :param filename: full filename of the L2Std file to be read
    :type filename: string
    :returns: set of variables and their values as specified by the user
    :rtype: dict
    '''
    h5File = h5py.File(filename, 'r')
```

```
data = {}

for name in VARIABLES:

    try:
        h5File[VARIABLES[name]]
    except KeyError:
        print "Bad variable path for '%s', reading data without it" % name
        continue

    thisVariable = h5File[VARIABLES[name]]
    data.update({ name : thisVariable.value })

return data
```

10 Contact Information

For the most up-to-date contact information, please refer to oco2.jpl.nasa.gov.

11 Acknowledgements and References

11.1 Acknowledgements

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

11.2 Additional Resources

There are a number of other project documents that the user should refer to as they work with the data

1. L1B ATBD – This Algorithm Theoretical Basis Document document describes the process used to take the un-calibrated spectrum to calibrated radiance spectra.
2. L2 ATBD – This ATBD steps through the physics and implementation of the level 2 algorithm.
3. ATBD for IMAP-DOAS and ABO2 – These ATBD documents describe the two methods of identifying potentially cloudy footprints, in what we refer to as the prescreening step. These data are then used for setting data quality and data selection levels.
4. Published papers – In addition to the references in the next section, there are a number of published papers describing the algorithm, application to GOSAT, prescreening steps, etc. Please see the most up to date list of publications on oco2.jpl.nasa.gov

11.3 References

11.3.1.1 Links

- http://oco2.jpl.nasa.gov

Level 2 algorithm information

- ACOS Level 2 Algorithm Theoretical Basis Document, JPL D-65488

Releases and publications

- http://www.jaxa.jp/press/2009/02/20090209_ibuki_e.html

11.3.1.2 OCO-2 Mission

- Canadell, J.G., C. Le Quere, M.R. Raupach, C.B. Field, E.T. Buitenhuis, P. Ciais, T.J. Conway, N.P. Gillett, R.A. Houghton, and G. Marlan (2007), Contributions to accelerating atmospheric CO2 growth from economic activity, carbon intensity, and efficiency of natural sinks, *Proceedings of the National Academy of Sciences*, 47, 18866-18870.

- Crisp, D., R.M. Atlas, F.-M. Breon, L.R. Brown, J.P. Burrows, P. Ciais, B.J. Connor, S.C. Doney, I.Y. Fung, D.J. Jacob, C.E. Miller, D. O'Brien, S. Pawson, J.T. Randerson, P. Rayner, R.J. Salawitch, S.P. Sander, B. Sen, G.L. Stephens, P.P. Tans, G.C. Toon, P.O. Wennberg, S.C. Wofsy, Y.L. Yung, Z. Kuang, B. Chudasama, G. Sprague, B. Weiss, R. Pollock, D. Kenyon, S. Schroll (2004), The Orbiting Carbon Observatory (OCO) mission, *Advances in Space Research* 34 700–709.

- Crisp, D., C. E. Miller, and P. L. DeCola (2008), NASA Orbiting Carbon Observatory: Measuring the column averaged carbon dioxide mole fraction from space, *JARS*.

- Frankenberg, C., Chris O'Dell, Joseph Berry, Luis Guanter, Joanna Joiner, Philipp Köhler, Randy Pollock, Thomas E. Taylor (2014), Prospects for chlorophyll fluorescence remote sensing from the Orbiting Carbon Observatory-2, *Remote Sensing of Environment*, 147, 1-12.

- P.J. Rayner and D.M. O'Brien, The utility of remotely sensed CO2 concentration data in surface source inversions, *Geophys. Res. Lett*. 28, 175, (2001).

11.3.1.3 Algorithms and Retrievals

- Crisp, D. B. M. Fisher, C. O'Dell, C. Frankenberg, R. Basilio, H. Bosch, L. R. Brown, R. Castano, B. Connor, N. M. Deutscher, A. Eldering, D. Griffith, M. Gunson, A. Kuze, L. Mandrake, J. McDuffie, J. Messerschmidt, C. E. Miller, I. Morino, V. Natraj, J. Notholt, D. M. O'Brien, F. Oyafuso, I. Polonsky, J. Robinson, R. Salawitch, V. Sherlock, M. Smyth, H. Suto, T. E. Taylor, D. R. Thompson, P. O. Wennberg, D. Wunch and Y. L. Yung (2012), The ACOS CO2 retrieval algorithm – Part II: Global XCO2 data characterization, *Atmos. Meas. Tech. Discuss*., 5, 687-707.
- Frankenberg, C., Platt, U., and Wagner, T. (2005), Iterative maximum a posteriori (IMAP-) DOAS for retrieval of strongly absorbing trace gases: Model studies for CH4 and CO2 retrieval from near- infrared spectra of SCIAMACHY onboard ENVISAT, *Atmos. Chem. Phys.*, 5, 9–22.
- O'Dell, C.W., B. Connor, H. Boesch, D. O'Brien, C. Frankenberg, R. Castano, M. Christi, D. Crisp, A. Eldering, B. Fisher, M. Gunson, J. McDuffie, G. C. Toon, P. O. Wennberg, and D. Wunch (2011), The ACOS CO2 retrieval algorithm – Part I: Description and validation against synthetic observations, *Atmos. Meas. Tech. Discuss., 4*, 6097–6158, 2011.
- Mandrake, L., C. Frankenberg, C. W. O'Dell, G. Osterman, P. Wennberg and D. Wunch (2013), Semi-autonomous sounding selection for OCO-2, *Atmos. Meas. Tech., 6*, 2851-2864.
- Taylor, T.E., C. O'Dell, D.M. O'Brien, N. Kikuchi, T. Yokota, T. Nakajima, H. Ishida, D. Crisp, and T. Nakajima (2011), Comparison of cloud screening methods applied to GOSAT near-infrared spectra, IEEE *Trans. Geosci. Rem. Sens*., doi: 10.1109/TGRS.2011.2160270.
- Rodgers, C. (2000) Inverse Methods for Atmospheric Sounding: Theory and Practice. World Scientific Publishing Co Pte Ltd.

11.3.1.4 Chlorophyll Fluorescence

- Frankenberg, C., Fisher, J., Worden, J., Badgley, G., Saatchi, S., Lee, J.-E., et al. (2011), New global observations of the terrestrial carbon cycle from GOSAT: Patterns of plant fluorescence with gross primary productivity. *Geophysical Research Letters*, 38(17), L17706.
- Frankenberg, C., O'Dell, C., Guanter, L., & McDuffie, J. (2012), Remote sensing of near-infrared chlorophyll fluorescence from space in scattering atmospheres: implications for its retrieval and interferences with atmospheric CO2 retrievals. *Atmospheric Measurement Techniques*, 5(8), 2081–2094, doi: 10.5194/amt-5-2081-2012.
- Joiner, J., Y. Yoshida, A. P. Vasilkov, Y. Yoshida3 , L. A. Corp4 , and E. M. Middleton (2011), First observations of global and seasonal terrestrial chlorophyll fluorescence from space, *Biogeosciences*, 8, 637–651.

11.3.1.5 Validation

Papers related to validation of the ACOS data product, plans for OCO-2 data validation or the TCCON network:

- Deutscher et al. (2010), Total column CO_2 measurements at Darwin, Australia - site description and calibration against in situ aircraft profiles, *Atmos. Meas. Tech., 3*, 947–958, doi: 10.5194/amt-3-947-2010.
- Keppel-Aleks et al. (2011), Sources of variations in total column carbon dioxide, *Atmospheric Chemistry and Physics, 11,* 3581–3593, doi: 10.5194/acp-11-3581-2011, http://www.atmos-chem-phys.net/11/3581/2011.
- Maxwell, S., R. Lee., H.R. Pollock (2014), Preflight Radiometric Calibration of Orbiting Carbon Observatory 2, in preparation.
- Messerschmidt et al. (2011), Calibration of TCCON column-averaged CO_2: the first aircraft campaign over European TCCON sites, *Atmos. Chem. Phys. Discuss., 11,* 14 541-14 582, doi: 10.5194/acpd-11-14541-2011.
- Washenfelder et al. (2006), Carbondioxide column abundances at the Wisconsin Tall Tower site, *J. Geophys. Res*, 111, D22305, doi: 10.1029/2006JD007154.
- Wunch et al. (2010), Calibration of the Total Carbon Column Observing Network using aircraft profile data, *Atmospheric Measurement Techniques*, 3, 1351–1362, doi: 10.5194/amt-3-1351-2010, http://www.atmos-meas-tech.net/3/1351/2010.
- Wunch et al. (2011a), The Total Carbon Column Observing Network, *Phil. Trans. R. Soc. A*, 369, 2087–2112, doi: 10.1098/rsta.2010.0240.
- Wunch et al. (2011b), A method for evaluating bias in global measurements of CO_2 total columns from Space, *Atmos. Chem. Phys. Discuss., 11,* 20899-20946.

12 Acronyms

ABO2	O_2-A Band cloud screening algorithm
ABSCO	Absorption Coefficients
ACOS	Atmospheric CO_2 Observations from Space
ACS	Attitude and Control System
AFE	Analog Front-end Electronics
aod	Aerosol Optical Depth
ARP	Ancillary Radiometric Product
ATBD	Algorithm Theoretical Basis Document
CO_2	Carbon Dioxide
CSU	Colorado State University
DEM	Digital Elevation Model
DOAS	Differential Optical Absorption Spectroscopy
ECMWF	European Centre for Medium-Range Weather Forecasts
ECS	EOSDIS Core System
EOS	Earth Observing System
EOS A-Train	EOS afternoon constellation
EOSDIS	Earth Observing System Data and Information System
ESMO	Earth Science Mission Operations
FP	Full Physics (algorithm)
FPA	Focal Plane Array (or Assembly)
FWHM	Full Width at Half Maximum
GES DISC	Goddard Earth Sciences Data and Information Services Center
GOSAT	Greenhouse Gases Observing Satellite
GSFC	Goddard Space Flight Center
H_2O	Water
HDF	Hierarchical Data Format
IFOV	Instantaneous Field of View
ILS	Instrument Line Shape
IMAP-DOAS	Iterative Maximum a Posteriori Differential Optical Absorption Spectroscopy
L	(0,1,..) Level 0, 1, etc. data product
LOS	Line of Sight
NASA	National Aeronautics and Space Administration
NCSA	National Center for Supercomputing Applications
O_2	Oxygen
OCO	Orbiting Carbon Observatory
OCO-2	Orbiting Carbon Observatory-2
PGE	Product Generation Executive
SC	Spacecraft
SCF	Science Computing Facility
SDOS	Science Data Operations System
SIF	solar-induced chlorophyll fluorescence
SNR	Signal to Noise Ratio
SRTM	Shuttle Radar Topography Mission
TAI	International Atomic Time
TCCON	Total Carbon Column Observing Network
UTC	Coordinated Universal Time
VCD	Vertical Column Density

VMR	Volume Mixing Ratio
WMO	World Meteorological Organization
WRS-2	Worldwide Reference System-2
X_{CO2}	Column-averaged dry air mole fraction of atmospheric CO_2

13 Appendix A

13.1 read_h5_files.pro

```
; FUNCTION read_h5_files, file, reader,$
;          _extra=_extra, verbose=verbose, $
;           NN=NN, counttotal=counttotal
;
; PURPOSE
;   This function reads part or all of multiple HDF-5 files.
;   It essentially calls "read_h5_file" for each input file,
;   then packages everything together.
;
;
; INPUTS
;   FILE: The input hdf-5 file name
;
;   READER: [optional] Name of the Function to read each file.  Default is "read_h5_file".
;
; KEYWORD INPUTS:
;   VERBOSE: Set this to turn verbose output on.  Default is off.
;
;   _EXTRA:   Any input keywords accepted by the READER.  If you use "read_h5_file", the
default,
;           see the code header for its accepted keywords.
;
;   NN:  Set this to the maximum number of data objects (ie soundings) per file.
;       It will try to guess, but it sometimes is too low.
;
; KEYWORD OUTPUTS:
;   COUNTTOAL:  This returns the total number of data objects (e.g. soundings) read across all
input files.
;

; Written by Chris O'Dell
; Colorado State University
; March 2014
;
; This work is not copyrighted and may be shared or modified without the direct
; consent of the author. It is provided on an 'as-is' basis and may include bugs,
; errors, etc.
;
; Please direct any questions to christopher.odell@colostate.edu
;
FUNCTION read_h5_files, files, reader, _extra=_extra, verbose=verbose, NN=NN,
counttotal=counttotal

; NN : max # of items per file. used in sizing output array
```

```
  ; this reads multiple h5 files.  it can use any single-file reader you like.  typically
"read_h5_file" is used.


  ON_ERROR, 2

  if n_elements(reader) eq 0 then reader = 'read_h5_file'
  nfiles = n_elements(files)

  f = 0L
  n = 0L
  CATCH, error_status
  IF Error_status NE 0 THEN BEGIN
    print, 'READ_H5_Files: Error in concatenating the two objects.'
    print, 'Try removing fields that are different sizes (Metadata is a common culprint).'
    print, 'The main data object looks like:'
    help, all[0]
    print, 'The trouble-causing data ojbect looks like:'
    help, d[0]
    PRINT, 'Error message: ', !ERROR_STATE.MSG
    return,-1
      Catch, /cancel
  ENDIF
  repeat begin
      if f eq 0 then begin
          repeat begin
              d = CALL_FUNCTION(reader, files[f], _extra=_extra, count=nd)
              if keyword_set(verbose) then print, f, nd
              f+=1
          endrep until nd GT 0
          elt_ = d[0]
          if n_elements(NN) eq 0 then begin
              ; must estimate the mean # of entries per file
              ; assume it scales with the file sizes
              f_info = file_info(files)
              NN = long(nd * total(f_info.size) / (float(f_info[f-1].size)*nfiles) * 2.5)
              if keyword_set(verbose) then print, 'Assuming approximately ' + sc(NN) + ' entries
per file on average.'
          endif
          all = replicate(elt_, nfiles*NN)
          all[0:nd-1] = d
          n += nd
      endif else begin
          d = CALL_FUNCTION(reader, files[f], _extra=_extra, count=nd)
          if nd GT 0 then begin
              all[n:n+nd-1] = d
              n += nd
          endif
```

```
          if keyword_set(verbose) then print, sc(f)+'/' + sc(nfiles) + ': ' +
file_basename(files[f]) + ': ' + sc(nd)+ ': ' + sc(n)
          f += 1
      endelse
  endrep until f GE nfiles
  counttotal = n
  if n eq 0 then return, -1

  all = all[0:n-1]
  return, all

  END
```

13.2 read_h5_file.pro

```
; FUNCTION read_h5_file, file, root, verbose=verbose, reform=reform, $
;                        success=success, count=count, $
;                        readlist=readlist, skiplist=skiplist, $
;                        condense=condense
;
; PURPOSE
;   This function reads part or all of an HDF-5 file.
;   The default is to read all of it, but it is quite versatile in reading
;   only some of the fields within a file.  It puts everything in into
;   a single output struct object.  It can also return an array of structs
;   if the CONDENSE keyword is set.
;
; INPUTS
;   File: The input hdf-5 file name
;   root: [optional] The group name within the file to work under.  Useful if you only want
;         fields from within a specific group in the file.
;
; KEYWORD INPUTS:
;   VERBOSE:
;
;   REFORM: Set this to reform dimensions of value 1.  E.g. a field with size, [1,3,2] would be
returned
;           as simply [3,2].  This is ON by default.
;
;   CONDENSE:   Set this keyword to return an array of structs rather than a single structure.
What
;           this does is brings dimensions common to all read fields out of the fields
themselves,
;           and puts them at the level of the struct.  E.g. this lets you get an array of
soundings,
;           where each sounding has a latitude, longitide, time, etc.  Very useful in organizing
data.
;           ON by default.
;
```

```
;   SUCCESS:    Set this to 1 if the file was successfully read. 0 otherwise.
;
;   COUNT:  Returns the number of data objects read.   Will only be greater than 1 if CONDENSE is
set.
;
;   READLIST:   A string array containing fields or expressions of fields to read.  Wildcards in
the group
;               or field names are completely acceptable.  Default is to read all of the fields.
;
;   SKIPLIST:   A string array containing fields or expressions of fields to NOT read.  Wildcards
in the group
;               or field names are completely acceptable. Items in SKIPLIST take precedence over
items in READLIST.
;
;
;   Example Call:
;
;               For example you may call read_h5_file as follows:
;   IDL> data = read_h5_file('my_file.h5', READLIST = ['Sounding/*', 'Retrieval/xco2'],
SKIPLIST='Sounding/sounding_solar_zenith')
;
;       In this case, all fields within group Sounding are read, except for
sounding_solar_zenith.  The field Retrieval/xco2
;       would also be read.
;

; Written by Chris O'Dell
; Colorado State University
; March 2014
;
; This work is not copyrighted and may be shared or modified without the direct
; consent of the author. It is provided on an 'as-is' basis and may include bugs,
; errors, etc.
;
; Please direct any questions to christopher.odell@colostate.edu
;
FUNCTION switch_bad_char, name

    name = strswitch(name, ', ', '_')
    name = strswitch(name, '-','_')
    name = strswitch(name, ' ', '_')

    name = strswitch(name, string(195b), '')
    name = strswitch(name, string(177b), 'n')
    name = strswitch(name, string(164b), 'a')
    name = strswitch(name, string(133b), 'a')
    return, name
END
```

```
FUNCTION match_h5_names, select_list, full_list

; for each element in full_list, determine if there is a match with any element in select_list
nfull = n_elements(full_list)
nselect = n_elements(select_list)
match = bytarr(nfull)
full_list_low =strlowcase(full_list)
full_len = strlen(full_list)
for i = 0, nselect-1 do begin
    this = strlowcase(select_list[i])
    ; determine if this select field has a wildcard in it
    ; Wildcard Function only works if group names are included (or as */)
    wild= strpos(this, '*') GE 0 or strpos(this, '?') GE 0 $
        OR (strpos(this, '[') GE 0 AND strpos(this, ']') GE (strpos(this, '[')+2))
    if ~wild then begin
        this_len =strlen(this)
        pos =strpos(full_list_low, this)
        w = where(pos GE 0)
        if w[0] NE -1 then match[w] = match[w] OR (pos[w] EQ (full_len[w]-this_len)) ; this last
one requires the names to match exactly, but not groups
    endif else begin
        w = where(strmatch(full_list_low, this))
        if w[0] NE -1 then match[w] = 1b
    endelse
endfor

return, match

END

FUNCTION read_h5_group_recursive, fid, group_name, dlist, got_data=got_data, $
                    quiet=quiet,reform=reform, condense=condense

    ; dlist are the fully-qualified fields to read

    quiet = keyword_set(quiet)
    if n_elements(condense) eq 0 then condense=1
    ngroup = h5g_get_nmembers(fid, group_name)
    got_data=0b
    for i = 0, ngroup-1 do begin
        member_name = h5g_get_member_name(fid, group_name, i)
        if group_name ne '/' then full_name = group_name + '/'+ member_name else
full_name=member_name
;        print, member_name
        info = h5g_get_objinfo(fid, full_name)
        success=0b
        CASE info.type of
        'GROUP' :   begin
                        if ~keyword_set(quiet) then print, 'Reading group ' + full_name
```

```
                        data = read_h5_group_recursive(fid, full_name, dlist, got_data=success, $
                                                       condense=condense,
quiet=quiet,reform=reform)
                    end
        'DATASET' : begin
                        if elt(full_name, dlist) then begin ; tag for getting or not getting a
data set
                            dat_id = h5d_open(fid, full_name)
                            storage_size = h5d_get_storage_size(dat_id)
                            if storage_size LE 0 then begin
                                if ~keyword_set(quiet) then print, 'Dataset ' + full_name + ' has
storage size = 0.'
                            endif else begin
                                if ~quiet then print, 'Reading ' + full_name
                                data = h5d_read(dat_id) ; read it all (no count or start or
stride)

                                if keyword_set(reform) then data=reform(data)
                                success=1b
                                ; add this type to the current running output struct
                            endelse
                            h5d_close, dat_id
                        endif
                    end
        endcase
        if success then begin ; if successfully got this field (or group), add it to the output
struct
            if n_elements(out) GT 0 $
                then out = create_struct(out, switch_bad_char(member_name), data) $
                else out = create_struct(switch_bad_char(member_name), data)
            got_data=1b
        endif
    endfor



    ; look for commong # of entries in last dimension and combine

    if got_data then begin
        if n_tags(out) eq 1 then begin
            if size(out.(0), /type) eq 8 then out = out.(0) else begin
                if condense GT 0 then repackage_struct, out, Nremove=condense
            endelse
        endif else begin
            if condense GT 0 then repackage_struct, out, Nremove=condense
        endelse
        if keyword_set(reform) then out = reform(out)
        return, out
    endif
```

```
    return, -1

END

FUNCTION read_h5_file, file, root, verbose=verbose, reform=reform, $
                       success=success, count=count, $
                       readlist=readlist, skiplist=skiplist, $
                       condense=condense

    filepath = file_search(file, /fully_qual, count=found)
    quiet=1b-keyword_set(verbose)
    if n_elements(condense) eq 0 then condense=1
    if n_elements(reform) eq 0 then reform=1
    if ~found then begin
        print, 'File ' + file + ' does not exist.'
        return, -1
    endif
    if ~h5f_is_hdf5(filepath) then begin
        print, 'File ' + file + ' is not an HDF5 file.'
        return, -1
    endif

    list = h5_list_datasets(file) ; if this fails then we are hosed
    if size(list, /type) LT 7 then begin
        print, 'Could not access h5 dataset names within ' + file + '.'
        return, -1
    endif
    nvar = n_elements(list)
    if n_elements(skiplist) GT 0 OR n_elements(readlist) GT 0 then begin
        if n_elements(skiplist) GT 0 then skip = match_h5_names(skiplist,list) else skip =
bytarr(nvar)+0b
        if n_elements(readlist) GT 0 then keep = match_h5_names(readlist,list) else keep =
bytarr(nvar)+1b
        wkeep = where(keep AND ~skip, nvar)
        if wkeep[0] eq -1 then begin
            print, 'No kept variables in h5 file. Returning -1.'
            return, -1
        endif
        list = list[wkeep]
    endif

    fid = h5f_open(filepath)
    if n_elements(root) eq 0 then root = '/'
    all = read_h5_group_recursive(fid, root, list, quiet=quiet, got_data=success,$
                                  reform=reform, condense=condense)
    count=0
    if success then count=n_elements(all)
    h5f_close, fid
```

```
    return, all

END
```

13.3 read_h5_field.pro

```
; FUNCTION read_h5_field, file, varname, $
;                         start=start, count=count, $
;                         success=success, check=check, $
;                         quiet=quiet, no_data=no_data
;
; PURPOSE
;   This function reads data from a single variable (or field) within an HDF-5 file.
;   It has a number of options that will modify its behavior.
;
; REQUIED INPUTS
;   File: The input hdf-5 file name
;
;   varname:
;
; OPTIONAL INPUT KEYWORDS
;   start:  An optional vector containing the starting position to read. The default start
position is [0, 0, ...].
;
;   count:  An optional vector containing the counts to be used in reading the field.
;           COUNT is a 1-based vector with an element for each dimension of the data to be
written.
;           Note that counts do not have to match the dimensions of the data field to be read.
;           The default count vector is the dimensionality of Value.
;
;   check:  Use this keyword if you roughly know the field name but aren't perfectly sure.
;           The program will look for the best match with what you typed.  If no match
;           can be found, an error will result.  Good if you know the first part of the field
name,
;           but not necessarily the group it lives in, or even the entire field name.
;
;   quiet:  Set this to not ever print anything out when called.
;
; OPTIONAL OUTPUT KEYWORDS
;   success:  Set to 1 if the file was successfully read. 0 otherwise.
;   no_data:  Set to 1 if the file contained no data (though it might exist). 0 otherwise.
;
; Written by Chris O'Dell
; Colorado State University
; March 2014
;
; This work is not copyrighted and may be shared or modified without the direct
; consent of the author. It is provided on an 'as-is' basis and may include bugs,
; errors, etc.
```

```
;
; Please direct any questions to christopher.odell@colostate.edu
;
function read_h5_field, file, varname, start=start, count=count, success=success, check=check,
quiet=quiet, no_data=no_data

    ; file : name of hdf-5 file
    ; varname: fully-qualified path to variable within hdf-5 file
    ;          [e.g., 'FrameSampleMeasurements/sample_measurements_weak_co2']

    COMMON read_h5_field_common, last_file_read, dlist

    success = 0
    no_data=0
    filepath = file_search(file, /fully_qual, count=found)

    if ~found then begin
        if ~keyword_set(quiet) then print, 'File ' + file + ' does not exist.'
        return, -1
    endif

    if ~h5f_is_hdf5(filepath) then begin
        if ~keyword_set(quiet) then print, 'File ' + file + ' is not an HDF5 file.'
        return, -1
    endif

    ; Find varname within this file.
    ; Use common block so we're not continually reading the dataset list
    if keyword_set(check) then begin
        if n_elements(last_file_read) eq 0 then dlist = h5_list_datasets(filepath) else begin
            if last_file_read ne filepath then dlist = h5_list_datasets(filepath)
        endelse
        last_file_read = filepath
        if n_elements(dlist) eq 0 then begin
            if ~keyword_set(quiet) then print, 'There are no datasets in this file!.'
            return, -1
        endif
        w = where(varname eq dlist)
        if w[0] eq -1 then begin
            w = where(strlowcase(varname) eq strlowcase(dlist))
            if w[0] eq -1 then begin
                p = strpos(strlowcase(dlist), strlowcase(varname))
                w = where(p GE 0)
                if w[0] eq -1 then begin
                    if ~keyword_set(quiet) then print, 'Dataset ' + varname + ' not found within
this file!!'
                    return, -1
                endif else true_varname = dlist[w[0]]
            endif else true_varname = dlist[w[0]]
```

```
        endif else true_varname = varname

        if true_varname ne varname AND ~keyword_set(quiet) then begin
            print, varname + ' not found.  Reading existing dataset ' + true_varname + '.'
        endif
    endif else true_varname = varname

    success = 1
    loc_id = h5f_open(filepath)
    dat_id = h5d_open(loc_id, true_varname)
    storage_size = h5d_get_storage_size(dat_id)
    if storage_size LE 0 then begin
        if ~keyword_set(quiet) then print, 'Dataset ' + true_varname + ' has storage size = 0.'
        success=1
        no_data=1
        return, -1
    endif

    if n_elements(start) GT 0 then begin
        datspace_id = h5d_get_space(dat_id)
        h5s_select_hyperslab,datspace_id, start, count, /reset
        memspace = h5s_create_simple(count)
    endif
    data = h5d_read(dat_id, file_space=datspace_id, mem=memspace)
    h5d_close, dat_id
    h5f_close, loc_id

    return, data

end
```

13.4 repackage_struct.pro

```
; FUNCTION repackage_struct, A, Nremove=Nremove
;
; A is a structure
; it can have any number of elements
; keep removing last dimensions when that dimension matches for all fields in A
;
; This FUNCTION is primarily used by READ_H5_FILE and READ_H5_FILES.
;
; Written by Chris O'Dell
; Colorado State University
; March 2014
;
; This work is not copyrighted and may be shared or modified without the direct
; consent of the author. It is provided on an 'as-is' basis and may include bugs,
; errors, etc.
;
```

```
; Please direct any questions to christopher.odell@colostate.edu
;
FUNCTION remove_dims, x, w

    ; x is an array, potentially multi-dimensional
    ; remove the dimensions with index w and return it

    xdims = size(x,/dim)
    nx = n_elements(xdims) ; number of dimensions in x
    if xdims[0] eq 0 then nx = 0 ; x is a scalar

    N = n_elements(w) ; number of dimensions to remove
    if nx LT N then begin
        print, 'error!: X Does not have enough dimensions!'
        STOP
    endif

    command = 'out=x['
    for d = 0, nx-1 do begin
        if elt(d, w) then command += '0' else command += '*'
        if d eq (nx-1) then command += ']' else command += ','
    endfor
    dummy = execute(command)
 ;   out = reform(out)  ; hmmm, this is causing the problem.  kill it?
    if n_elements(out) eq 1 then out = out[0]
    return, out
END

PRO repackage_struct, A, Nremove=Nremove

    ; A is a structure
    ; it can have any number of elements
    ; keep removing last dimensions when that dimension matches for all fields in A


    nA = n_elements(A)
    names = tag_names(A)
    ; 1 determine number of matching dimensions on the backside
    ntags = n_tags(A)
    match_dims = bytarr(10) + 1b ; 10 is the maximum number of dimensions to check
    first_dims = lonarr(10)
    for t=0,ntags-1 do begin
        field = A[0].(t)
        D = reverse(size(field, /dim))
        nD = n_elements(D)
        if D[0] eq 0 then nD = 0
        if nD GT 0 then begin
            if t eq 0 then first_dims[0:nd-1] = D $
                      else match_dims[0:nd-1] = match_dims[0:nd-1] AND first_dims[0:nd-1] EQ D
```

```
        endif
        if nd LT 10 then match_dims[nd:*] = 0b
    endfor
    ; now count up the number of matching dimensions
    nmatch = 0
    for d = 0, 9 do if match_dims[d] then nmatch += 1 else break

    if nmatch EQ 0 then return

    ; 2. Repackage the structure
    if keyword_set(Nremove) then Nr=Nremove else Nr=10
    if nmatch GT Nr then nmatch=nr
    match_dims = reverse(first_dims[0:nmatch-1])

    ; 2a.  Create the basic elemnent of B
     for t = 0, ntags-1 do begin
        this = A[0].(t)
        this_dims = size(this, /dim)
        ndims_this = n_elements(this_dims)
        dims_to_remove = bytarr(ndims_this)
        dims_to_remove[ndims_this-nmatch:*] = 1b
        whdims =where(dims_to_remove)
        this2 = remove_dims(this, whdims)
        if t eq 0 then Belt_ = create_struct(names[t], this2) $
                  else Belt_ = create_struct(Belt_, names[t], this2)
    endfor

    Bdims = match_dims
    if nA GT 1 then Bdims = [Bdims, size(A,/dim)]
    B = replicate(Belt_, Bdims)
    for t = 0, ntags-1 do B.(t) = A.(t)
    A = temporary(B)

END
```

13.5 H5_list_datasets.pro

```
; FUNCTION h5_list_datasets, file , root
;
; PURPOSE
;   This function lists all the variables in an HDF-5 file.
;   It can also be used to list the variables within a given group in a file.
;
; INPUTS
;   File: The input hdf-5 file name
;   root: [optional] The group name within the file to list under.
;
; Written by Chris O'Dell
; Colorado State University
```

```
; March 2014
;
; This work is not copyrighted and may be shared or modified without the direct
; consent of the author. It is provided on an 'as-is' basis and may include bugs,
; errors, etc.
;
; Please direct any questions to christopher.odell@colostate.edu
;
PRO h5_object_list, fid, group_name, dlist

    ngroup = h5g_get_nmembers(fid, group_name)
    for i = 0, ngroup-1 do begin
        member_name = h5g_get_member_name(fid, group_name, i)
        if group_name ne '/' then member_name = group_name + '/'+ member_name
;         print, member_name
        info = h5g_get_objinfo(fid, member_name)
        CASE info.type of
        'GROUP' :  h5_object_list, fid, member_name, dlist
        'DATASET' : if n_elements(dlist) eq 0 then dlist = member_name else dlist = [dlist,
member_name]
        endcase
    endfor

END

FUNCTION h5_list_datasets, file, root

    filepath = file_search(file, /fully_qual, count=found)

    if ~found then begin
        print, 'File ' + file + ' does not exist.'
        return, -1
    endif

    if ~h5f_is_hdf5(filepath) then begin
        print, 'File ' + file + ' is not an HDF5 file.'
        return, -1
    endif

    fid = h5f_open(filepath)
    if n_elements(root) eq 0 then root = '/'
    h5_object_list, fid, root, dlist
    h5f_close, fid

    if root ne '/' then begin
        ; remove the leading group name
        nroot = strlen(root)
        nd = n_elements(dlist)
        for i =0, nd-1 do dlist[i] = strmid(dlist[i], nroot+1, strlen(dlist[i])-nroot-1)
```

```
    endif

    return, dlist

END
```